

AIAA Flight Simulation Technologies Conference

A
COLLECTION
OF
TECHNICAL
PAPERS

Monterey, CA
August 9-11, 1993



For permission to copy or republish, contact:
American Institute of Aeronautics and Astronautics
The Aerospace Center
370 L'Enfant Promenade, SW
Washington, DC 20024-2518

AIAA/Aerospace America/ Flight International Flight Simulation Technologies Conference



General Chairperson
JAMES DAVIS
IVEX Corporation



Technical Program Chairperson
EDWARD M. BOOTHE

PHOTO
NOT
AVAILABLE

Administrative Chairperson
KAREN WALKER
Media Consultant

**Monterey Marriott, Doubletree at Fisherman's Wharf,
and the Monterey Conference Center
Monterey, California
August 9–11, 1993**

INTRODUCTORY REMARKS

This 1993 Flight Simulation Technologies Conference Collection of Technical Papers is a reflection of the efforts of a dedicated group of people who contributed their talents towards a successful conference and a valuable product. We are indeed appreciative of these efforts and proud of the result. As you review this Collection from time to time or use the volume for reference, we hope that you will find the work supportive of your needs.

The Collection includes over 60 papers in 14 sessions. More than forty percent of them have been contributed by authors from foreign lands including Brazil, Canada, China, Germany, The Netherlands and Russia. They include history, modeling, simulation use, and simulation technology. These papers also represent the state of flight simulation technology during a time when the aerospace, both civilian and military, are experiencing the forces of reduced budgets and austere operations. Simulation continues, however, to be a cost effective mechanism for training, mission rehearsal, testing, airman certification and even aircraft certification. In fact, simulation becomes more of an alternative as the use of real aircraft becomes less economically feasible for tasks that can be simulated.

There is considerable synergy between the papers of this Collection and those of the concurrent conferences on Guidance, Navigation and Control; Atmospheric Flight Mechanics; and Applied Aerodynamics. This synergism results because simulation quite easily and quite readily extends into other disciplines and becomes a tool for use in studying numerous phenomena that are not themselves within the science of simulation. Of course, without the study of other disciplines or the necessity of training, rehearsal, and testing there would be little reason to simulate. Hence, simulation has become an integral part of other sciences as well as becoming a science unto itself.

Appreciation is especially extended to the fifteen chairpersons of the conference who have been invaluable links in the chain of organization; the authors without whom the effort would be impossible; Ms. Bridget Supik, AIAA, for technical papers management and Ms. Julie Walker of the AIAA for the most professional and dedicated management of the conference.

James L. Davis, PhD
1993 Flight Simulation Technologies Conference
General Chairperson

Edward M. Boothe
Technical Program Chairperson

Ms. Karen Walker
Administrative Chairperson

Guidance, Navigation & Control Conference Atmospheric Flight Mechanics Conference Flight Simulation Technologies Conference Applied Aerodynamics Conference

Monday Morning / August 9, 1993							
9:00	9:30	10:00	10:30	11:00	11:30	12:00	12:30
8:00 am — 9:00 am Plenary — Contribution of Simulation to Airline Pilot Training CAPT WILLIAM H. TRAUB, VP Flight Standards and Training, United Airlines							Steinbeck Forum
Session 1-GNC-1 — Advanced Estimation Theory and Applications Chaired by: C. D'SOUZA, Wright Lab., Eglin AFB, FL							Bonsai I
AIAA-93-3700 Estimate Fusion for Lunar Rendezvous R. Carpenter and R. Bishop, Univ. of Texas, Austin, TX	AIAA-93-3701 Bayesian State Estimation for Tracking and Guidance Using the Bootstrap Filter N. Gordon and D. Salmood, Defense Research Agency, Farnborough, UK; C. Ewing, USAF, Eglin AFB, FL	AIAA-93-3702 Development of an Optimized LEB Filter and its Applications to INS/GPS Test Data C. Antonini, Univ. of Pretoria, Pretoria, South Africa	AIAA-93-3703 Kalman Filters in ADA Comparison Study T. Ferry, J. Barhorst, and M. Feldman, McDonnell Douglas Aerospace, St. Louis, MO	AIAA-93-3704 Multisensor Multitarget Mixture Reduction Algorithms for Tracking L. Pao Mitre Corp., Bedford, MA	AIAA-93-3705 A Note on Target Estimation with Haar Functions P. Hempel, Lockheed Missiles & Space Co., Sunnyvale, CA	AIAA-93-3706 Linear Programming Control of an Air-to-Air Missile Using a Model Obtained by Nonlinear Identification N. Nabaa, Univ. of Texas, Austin, TX; M. Eggmann and R. Longchamp, Swiss Federal Institute of Technology, Lausanne, Switzerland	
Session 2-GNC-2 — Computational Control and Dynamics I Chaired by: D. MACKISON, Univ. of Colorado, Boulder, CO							Bonsai II
AIAA-93-3707 A Truncation Error Model and its Application to the Accuracy Analysis of Constraint Violations S. Yoon, KIAI Korean Air, Seoul, Korea; R. Howe and D. Greenwood, Univ. of Michigan, Ann Arbor, MI	AIAA-93-3708 Design of Small Gain Controller via Iterative Eigenstructure Assignment Y. Kim and D. Kum, Seoul National Univ., Seoul, Korea; J. Jenkins, Texas A&M Univ., College Station, TX	AIAA-93-3709 A COMPACT Model Reduction Methodology for Articulated, Multiflexible Body Structures A. Yee, JPL, Pasadena, CA	AIAA-93-3710 Assumed Modes Method and Nonlinear Flexible Multibody Dynamics S. Tadikona and T. Moradin, Grumman SSD, Reston, VA	AIAA-93-3711 Deployment Control of a Cable Connecting a Ship to an Underwater Vehicle A. Banerjee and V. Do, Lockheed Missiles & Space Co., Palo Alto, CA	AIAA-93-3712 Simulation of the Assembly Dynamics and Control of the Space Station Freedom P. Cooper, J. Garrison, and R. Montgomery, NASA Langley, Hampton, VA; S.-C. Wu and A. Stockwell, Lockheed Engineering & Sciences Co., Hampton, VA; M. Demeo, Vigyan, Hampton, VA	AIAA-93-3713 Trajectory Optimization for a National Launch System Vehicle E. Chen, F. Boeltz, and J. Sullivan, Charles Stark Draper Lab., Cambridge, MA	
Session 3-GNC-3 — Spacecraft Dynamics Chaired by: P. STOLTZ, Orbital Sciences Corp., Boulder, CO							Bonsai III
AIAA-93-3714 Near Minimum Time Maneuvers of Large Space Structures Using Parameter Optimization M. Carter, T. Singh, and S. Vadali, Texas A&M Univ., College Station, TX	AIAA-93-3715 Retrieval Control of a Tethered Subsatellite under Effect of Tether Elasticity H. Fuji and K. Kokubun, Tokyo Metropolitan Institute of Technology, Tokyo, Japan	AIAA-93-3716 Precession and Circularization of Elliptical Space-Tether Motion J. Chapel, Martin Marietta Astronautics, Denver, CO; P. Grossrosde, Orbital Sciences Corp., Chandler, AZ	AIAA-93-3717 A Testbed for Research on Manipulator-Coupled Active Spacecraft R. Montgomery and J. Garrison, NASA Langley, Hampton, VA; P. Tobbe and J. Weathers, Logicon Control Dynamics, Hampton, VA; D. Ghosh, Lockheed Engineering & Sciences Co., Hampton, VA	AIAA-93-3718 Precision Attitude Control of a Space Structure Using a Three-Rigid Manipulator R. Mukherjee and M. Zurofski, Naval Postgraduate School, Monterey, CA	AIAA-93-3719 Output Zeroing Submanifold and Nonlinear Regulation of Space Station: A Geometric Approach S. Singh and A. Iyer, Univ. of Nevada, Las Vegas, NV	AIAA-93-3720 Hamiltonian Dynamics of a Rigid Body with Momentum Wheels in a Central Gravitational Field P.T. Chen, K.-Y. Lian and L.S. Wang, National Taiwan Univ., Taiwan, ROC	

Session 4-GNC-4—Advanced Methods for Flexible Structure Control <i>Chaired by:</i> A. BICOS, McDonnell Douglas Space Systems Co., Huntington Beach, CA						Redwood
AIAA-93-3721 Genetic Design of Active Controllers for Flexible Space Structures B. Porter and M. Borain, Univ. of Salford, Salford, UK	AIAA-93-3723 Methods for Controlling Plate Vibrations Using Piezoelectric Actuators E. Palangas, J. Dworak, and S. Koshigoe, Aerospace Corp., Los Angeles, CA	AIAA-93-3724 Slew Maneuver On/Off Thrusters and Multiple Feedback Switching Strategies B. Agrawal and H. Bang, Naval Postgraduate School, Monterey, CA	AIAA-93-3725 A Piezoelectric Transformer C. Won, Lockheed Engineering & Sciences Co., Hampton, VA	AIAA-93-3726 Design of the Reduced LQG Compensator for the Deep-Space Network Antennas W. Gawronski, JPL, Pasadena, CA	AIAA-93-3727 Determination of Spacecraft Liquid Fill Fraction A. Chailioner, Hughes Space & Communications Group, El Segundo, CA	
Session 5-GNC-5—AI Methods for Flight Control Law Synthesis <i>Chaired by:</i> W. BAKER, Charles Stark Daper Lab., Cambridge, MA						Ironwood
AIAA-93-3728 Control Augmentation System (CAS) Synthesis via Adaptation and Learning P. Milington and W. Baker, Draper Lab., Nassau Bay, TX	AIAA-93-3729 Hopfield Neural Network for Adaptive Control M. Meurs, R. Smith, P. Chandier, and M. Pachter, Wright Lab., Wright-Patterson AFB, OH	AIAA-93-3731 Multiple Radial Basis Function Networks in Modeling and Control H. Youssef, Lockheed Aeronautical Systems Co., Marietta, GA	AIAA-93-3732 Application of Intelligent Search Strategies to Robust Flight Control for Hypersonic Vehicles G. Chamlioff, Charles Stark Draper Lab., Cambridge, MA	AIAA-93-3733 H _∞ Control Law Design Using Gradient-Based Neural Networks K. Blankinship, Raytheon Co., Tewksbury, MA		
Session 6-GNC-6—Missile Autopilots and Control <i>Chaired by:</i> A. WILSON, Nichols Research Corp., Huntsville, AL						Cottonwood
AIAA-93-3735 Adaptive Autopilot Design for an Exoatmospheric Interceptor O. Lam, Coleman Research Co., Fairfax, VA; A. Carstens, Wright Lab., Eglin AFB, FL	AIAA-93-3736 Autopilot Design for the Advanced Kinetic Energy Missile (ADKEM) Boost Phase H. Gratt, KBM Corp., Huntsville, AL; W. McCowan, U.S. Army Missile Command, Redstone Arsenal, AL	AIAA-93-3738 Missile Autopilot Robustness to Real and Complex Uncertainties Using a Parameter Space Robustness Test K. Wise, McDonnell Douglas Aerospace, St. Louis, MO; S. Kundur, Southern Illinois Univ., Edwardsville, IL	AIAA-93-3739 Simultaneous Reaction Jet and Aerodynamic Control of Missile Systems M. Innocenti, Università di Pisa, Pisa, Italy; A. Thukral, Auburn Univ., Auburn, AL	AIAA-93-3741 Preliminary Pulse Motor Optimization for a Surface-to-Air Missile C. Phillips, Naval Surface Warfare Center, Dahlgren, VA		
Session 7-GNC-7—Optimal Control Techniques <i>Chaired by:</i> B. WIE, Arizona State Univ., Tempe, AZ						Colton III
AIAA-93-3742 Hodograph Analysis in Aircraft Trajectory Optimization E. Clift, Virginia Tech, Blacksburg, VA; H. Seywald, Analytical Mechanics Associates, Hampton, VA; R. Bless, Lockheed Engineering & Sciences Co., Hampton, VA	AIAA-93-3743 On the Existence of Touch Points for First-Order State Inequality Constraints H. Seywald, Analytical Mechanics Associates, Hampton, VA; E. Clift, Virginia Tech, Blacksburg, VA	AIAA-93-3744 Stabilized Continuation Method for Solving Optimal Control Problems H. Fujii and T. Ohtsuka, Tokyo Metropolitan Institute of Technology, Tokyo, Japan	AIAA-93-3745 A Technique for Improving Numerical Stability and Efficiency in Singular Control Problems R. Mehlinhorn, K. Lesch, and G. Sachs, Technische Universität München, München, Germany	AIAA-93-3746 The Generalized Legendre-Chebyshev Condition on State/Control Constrained Arcs H. Seywald, Analytical Mechanics Associates, Hampton, VA; E. Clift, Virginia Tech, Blacksburg, VA	AIAA-93-3747 State Constrained Booster Trajectory Solutions via Finite Elements and Shooting R. Bless, NASA Langley, Hampton, VA; D. Hodges, Georgia Institute of Technology, Atlanta, GA; H. Seywald, Analytical Mechanics Associates, Hampton, VA	

Monday Morning / August 9, 1993							
9:00	9:30	10:00	10:30	11:00	11:30	12:00	12:30
Session 8-AFM-1—Aircraft Stability and Control <i>Chaired by: F. GARRETT JR., Arizona State Univ., Tempe, AZ</i>							Ferrante II
AIAA-93-3615 Effects of Geometry, Static Stability, and Mass Distribution on the Tumbling Characteristics of Generic Plying-Wing Models C. Frenaux, D. Vairo, and R. Whipple, Lockheed Engineering & Sciences Co., Hampton, VA	AIAA-93-3616 Analysis of the Stability Characteristics for a High-Performance Aircraft G. Guglieri, <i>Centro Studi Dinamica Fluida</i> , Torino, Italy; F. Quagliotti, <i>Politecnico di Torino</i> , Torino, Italy	AIAA-93-3617 Calculation of Transonic Longitudinal and Lateral-Directional Characteristics of Aircraft by the Small Disturbance Theory M. Thacker and E. Lan, Univ. of Kansas, Lawrence, KS	AIAA-93-3618 An Advanced Parallel Rotorcraft Flight Simulation Model: Stability Characteristics and Handling Qualities S. Sarathy, <i>Georgia Institute of Technology</i> , Atlanta, GA; V. Murthy, <i>Syracuse Univ.</i> , Syracuse, NY	AIAA-93-3619 Validation of the Rotary Balance Technique for Predicting Pitch Damping W. Blake, <i>Wright Lab.</i> , <i>Wright-Patterson AFB</i> , OH	AIAA-93-3620 Estimation of Neutral and Maneuver Points of Aircraft by Dynamic Maneuvers S. Srinathkumar, P. Madhuranath, and G. Gopalratnam, <i>National Aeronautical Lab.</i> , Bangalore, India		
Session 9-AFM-2—Unsteady Aerodynamics I <i>Chaired by: R. CUMMINGS, California Polytechnic Univ., San Luis Obispo, CA</i>							Colton I and II
AIAA-93-3622 Unsteady Aerodynamics for Air Force Needs (Invited-1 hr) T. Weeks and K. Nagaraja, <i>Wright Lab.</i> , <i>Wright-Patterson AFB</i> , OH	AIAA-93-3623 Status of the Validation of High-Angle-of-Attack Nose-Down Pitch Control Margin Design Guidelines M. Ogburn and J. Foster, <i>NASA Langley</i> , Hampton VA; J. Pahl, <i>NASA Dryden</i> , Edwards, CA; J. Wilson, J. Lackey, <i>Naval Air Warfare Center</i> , Patuxent River, MD	AIAA-93-3624 Lateral Control at High Angles of Attack Using Pneumatic Blowing Through a Chined Forebody A. Arena Jr., <i>Oklahoma State Univ.</i> , Stillwater, OK; R. Nelson, <i>Univ. of Notre Dame</i> , Notre Dame, IN; L. Schilt, <i>NASA Ames</i> , Moffett Field, CA	AIAA-93-3625 Kinematics and Aerodynamics of the Velocity Vector Roll W. Durham, F. Lutze, and W. Mason, <i>Virginia Tech</i> , Blacksburg, VA	AIAA-93-3626 Unsteady Aerodynamic Models for Maneuvering Aircraft C.-C. Hu and E. Lan, <i>Univ. of Kansas</i> , Lawrence, KS	AIAA-93-3627 A Qualitative Assessment of Control Effectors on an Advanced Fighter Configuration M. Alexander, <i>Wright Lab.</i> , <i>Wright-Patterson AFB</i> , OH		
Session 10-AFM-3—Projectile/Missile Aerodynamics: CFD and Analysis <i>Chaired by: P. PLOSTINS, U.S. Army Research Lab., Aberdeen Proving Ground, MD</i>							Ferrante III
AIAA-93-3628 On Computing Vortex Asymmetries about Cones at Angle of Attack Using the Canonical Navier-Stokes Equations D. Dusing and P. Orkisz, <i>Univ. of Cincinnati</i> , Cincinnati, OH	AIAA-93-3629 Base Drag Prediction of Missile Configurations F. Moore and T. Hymer, <i>Naval Surface Warfare Center</i> , Dahlgren, VA; F. Wilcox, <i>NASA Langley</i> , Hampton, VA	AIAA-93-3630 Numerical Study of a Hollow-Nosed Interceptor Design A. Spring, <i>Wright Lab.</i> , <i>Wright-Patterson AFB</i> , OH; C. Smith, <i>CFD Research Corp.</i> , Huntsville, AL	AIAA-93-3631 Analysis of Missile Configurations with Wrap-Around Fins Using Computational Fluid Dynamics G. Abale and T. Cook, <i>Wright Lab.</i> , Eglin AFB, FL	AIAA-93-3632 Application of Chimera Technique to Projectiles in Relative Motion J. Sabu and C. Nieuwicz, <i>U.S. Army Research Lab.</i> , <i>Aberdeen Proving Ground</i> , MD	AIAA-93-3633 Development of an Accuracy Criteria for Body-on-Fin Carryover Interference K. Burns, <i>McDonnell Douglas Aerospace</i> , St. Louis, MO		
Session 11-AFM-4—System Identification and Parameter Estimation <i>Chaired by: J. BATTERSON, NASA Langley, Hampton, VA</i>							Ferrante I
AIAA-93-3634 Identification of a Full Subsonic Envelope Nonlinear Aerodynamic Model of the F-14 Aircraft T. Trankle, <i>Systems Control of Technology</i> , Palo Alto, CA; S. Bachner, <i>Naval Air Warfare Center</i> , Patuxent River, MD	AIAA-93-3635 Frequency-Domain Identification of Unstable Systems Using K-31 Aircraft Test Data J. Kaletka and K. Fu, <i>DLR</i> , Braunschweig, Germany	AIAA-93-3636 Nonlinear Aerodynamics Modeling Using Multivariate Orthogonal Functions E. Morelli, <i>NASA Langley</i> , Hampton, VA	AIAA-93-3637 Analytical Solution and Parameter Estimation of Projectile Dynamics S. Weiss and K.-F. Doherr, <i>DLR</i> , Braunschweig, Germany; H. Schilling, <i>Rheinmetall GmbH</i> , Ratingen, Germany	AIAA-93-3638 On the Use of Back Propagation with Feed-Forward Neural Networks for the Aerodynamic Estimation Problem R. Hess, <i>Manassas</i> , VA	AIAA-93-3639 Estimation of Aerodynamic Coefficients Using Neural Networks H. Youssel, <i>Lockheed Aeronautical Systems Co.</i> , Marietta, GA	AIAA-93-3640 On Parameter Estimation of an Aeroelastic Aircraft S. Rasingham, <i>IIT-Kanpur</i> , Kanpur, India; A. Gosh, <i>ARDE</i> , Pune, India	
Session 12-FST-1—Flight Simulation History <i>Chaired by: J. HUSSAR, CAE-Link Corp., Binghamton, NY</i>							Santa Monica
AIAA-93-3545 Evolution of Flight Simulation L. Allen, <i>CAE Electronics</i> , Montreal, Quebec, Canada	AIAA-93-3547 Lessons Learned from a Historical Review of Piloted Aircraft Simulators at NASA S. Anderson, <i>NASA Ames</i> , Moffett Field, CA; R. Morrison, <i>SYHE</i> , Moffett Field, CA	AIAA-93-3548 Flight Simulation: An Enduring System of Innovation R. Miller, <i>Hydro-Quebec</i> , Montreal, Canada					

Monday Morning / August 9, 1993

9:00	9:30	10:00	10:30	11:00	11:30	12:00	12:30
Session 13-FST-2 —Simulation Modeling I <i>Chaired by: R. CHEN, NASA Ames, Moffett Field, CA</i>						Los Angeles	
AIAA-93-3549 Development and Validation of a Blade Element Mathematical Model for the AH-64 Apache Helicopter H. Mansur, U.S. Army ATCOM, Moffett Field, CA	AIAA-93-3550 An Advanced Parallel Rotorcraft Flight Simulator Model: Parallel Implementation and Performance Analysis S. Sarathy, Georgia Institute of Technology, Atlanta, GA; V. Murthy, Syracuse Univ., Syracuse, NY	AIAA-93-3551 Methodology for Integration of Control Loaders in Aircraft Simulators A. Katz, K. Krishnakumar, and M. Schamie, Univ. of Alabama, Tuscaloosa, AL	AIAA-93-3552 Dynamic Simulation Fidelity Improvement Using Transfer Function State Extrapolation W. Bezdek and L. Moody, McDonnell Douglas Aerospace, St. Louis, MO	AIAA-93-3553 Aircraft Threat Modeling from Performance Data M. Anderson, Virginia Tech, Blacksburg, VA; D. Schab, Naval Training Systems Center, Orlando, FL	AIAA-93-3554 Universal Controls Loading System L. Zaychik, A. Predtechensky, V. Rodchenko, and Y. Yashin, TsAGI, Moscow, Russia		
Session 14-FST-3 —Simulation Facilities I <i>Chaired by: J. HOUCK, NASA Langley, Hampton, VA</i>						San Diego	
AIAA-93-3555 Development of a Graphical User Interface to Facilitate Development and Certification of a New Commercial Airplane J. Little, Boeing Commercial Airplane Group, Seattle, WA	AIAA-93-3556 D'S: The Delphins Display Design System E. Theunissen, Delft Univ. of Technology, Delft, The Netherlands	AIAA-93-3557 The Flight Simulation Technology for Integrated Flight/Trajectory Control System K. Wang, Hughes Missile Systems Co., Alta Loma, CA	AIAA-93-3558 A High Fidelity Video Delivery System for Realtime Simulation Research D. Wilkins, C. Roach, W. Cook, and H. Huynh, SYRE, Moffett Field, CA	AIAA-93-3559 A Hybrid Complex of the Aircraft Intellectualized Control Systems Simulation at the Stage of Their Research Projecting A. Yakovlev and V. Vasilets, International Academy of Engineering, Moscow, Russia			
Session 15-AA-1 —CFD: Unsteady Aerodynamics <i>Chaired by: E. LEE-RAUSCH, NASA Langley, Hampton, VA</i>						San Carlos I	
AIAA-93-3400 Adaptive Computations of Flow around a Delta Wing with Vortex Bursting D. Modiano, ICOMP, Brookpark, OH; E. Murman, MIT, Cambridge, MA	AIAA-93-3401 Numerical Simulation of Wing-Wall Junction Flow for a Pitching Wing R. Newsome, USAF Academy, Colorado Springs, CO	AIAA-93-3402 Unsteady Flow Computations for a Three-Dimensional Cavity with and without an Acoustic Suppression Device N. Suhs, Calspan Corp., Arnold AFB, TN	AIAA-93-3403 Dynamic Stall of an Oscillating Wing—Part I G. Srinivasan, J. Ekaterinaris, and W. McCroskey, NASA Ames, Moffett Field, CA	AIAA-93-3404 Inviscid CFD for Store Separation Using Unified Boundary Conditions S. Chakravarthy, K.Y. Szema, and S. Ramakrishnan, Rockwell International Science Center, Thousand Oaks, CA; R. Burman and R. Shultz, Naval Air Warfare Center, China Lake, CA	AIAA-93-3495 Navier-Stokes Prediction of a Delta Wing in Roll with Vortex Breakdown N. Chaderjian and L. Schif, NASA Ames, Moffett Field, CA		
Session 16-AA-2 —Forebody Vortex Aerodynamics <i>Chaired by: G. MALCOLM, Eidetics International, Torrance, CA</i>						San Carlos II	
Forebody Vortex Control—A Progress Review (Invited-1 hr) G. Malcolm, Eidetics International, Torrance, CA	AIAA-93-3406 Tangential Forebody Blowing-Yaw Control at High Alpha W. Crowther and N. Wood, Univ. of Bath, Bath, UK	AIAA-93-3407 Side-Force Control on a Forebody of Diamond Cross-Section Angles of Attack D. Rao and G. Sharma, Vigyan, Hampton, VA	AIAA-93-3408 Symmetry Breaking in Vortical Flows over Cones—Theory and Numerical Experiments P. Hartwich, Vigyan, Hampton, VA	AIAA-93-3409 Bluntness and Blowing for Flowfield Asymmetry Control on Slender Forebodies F. Roos and C. Magness, McDonnell Douglas Corp., St. Louis, MO			

Session 17-AA-3 —Rotor/Experimental Aerodynamics <i>Chaired by: J. RUTHERFORD, McDonnell Douglas Helicopter Co., Mesa, AZ</i>							San Carlos III
AIAA-93-3410 Effect of Ground and Ceiling Planes on Shape of Energized Vortex V. Rossow, NASA Ames, Moffett Field, CA	AIAA-93-3411 Applications of the Shadowgraph Flow Visualization Technique to a Full-Scale Helicopter in Hover and Forward Flight A. Swanson, Sterling Software, Moffett Field, CA	AIAA-93-3412 Low-Speed Wind-Tunnel Test Results of the Canard Rotor/Wing Concept S. Bass, T. Thompson, and J. Rutherford, McDonnell Douglas Helicopter Co., Mesa, AZ; S. Swanson, NASA Ames, Moffett Field, CA	AIAA-93-3413 A Critical Assessment of UH-60 Main Rotor Blade Airfoil Data J. Tolah, NASA Ames, Moffett Field, CA	AIAA-93-3414 Performance Results from a Test of an S-76 Rotor in the NASA Ames 80 x 120 ft Wind Tunnel P. Shinoda, NASA Ames, Moffett Field, CA; W. Johnson, Johnson Aeronautics, Palo Alto, CA	AIAA-93-3415 Tilt Rotor Wing Download Measurements from a Small-Scale, Semispan Model J. Light, NASA Ames, Moffett Field, CA		

Session 18-AA-4 —Drag Prediction/Transonic Aerodynamics <i>Chaired by: N. MALMUTH, Rockwell International Science Center, Thousand Oaks, CA</i>							San Carlos IV
AIAA-93-3416 Computations of Aerodynamic Drag for Turbulent Transonic Profiles with and without Spin J.-K. Fu, Chinese Air Force Academy, Taiwan, ROC; S.-M. Liang, National Cheng Kung Univ., Taiwan, ROC	AIAA-93-3417 An Improved Far-Field Drag Calculation Method for Nonlinear CFD Codes J. Sibrough, Brook Park, OH	AIAA-93-3418 CFD Drag Predictions for a Wide Body Transport Fuselage with Flight Test Verification J. Rhodes, McDonnell Douglas Aerospace, St. Louis, MO; K. Ravindra, Parks College, Cahokia, IL; D. Friedman, Douglas Aircraft, Long Beach, CA	AIAA-93-3419 Computational Analysis of Drag Reduction and Buffet Alleviation in Viscous Transonic Flow over Porous Airfoils M. Gillan, Queen's Univ. Belfast, Belfast, UK	AIAA-93-3420 A Method for the Prediction of Induced Drag for Planar and Nonplanar Wings M. Maughmer, Pennsylvania State Univ., University Park, PA; K. Mortara, Lockheed, Marietta, GA	AIAA-93-3421 CFD Optimization of a Theoretical Minimum Drag Body S. Cheung, P. Aaronson, and T. Edwards, MCAT Institute, Moffett Field, CA		

Monday Afternoon / August 9, 1993							
2:00	2:30	3:00	3:30	4:00	4:30	5:00	5:30

Session 19-GNC-8—Guidance and Optimal Control Theory <i>Chaired by: B. JONES, AFIT, Wright-Patterson AFB, OH</i>							Bonsai I
AIAA-93-3748 Near Optimal, Asymptotic Tracking in Control Problems Involving State-Variable Inequality Constraints N. Markopoulos and A. Calise, Georgia Institute of Technology, Atlanta, GA	AIAA-93-3749 A Preconditioning Theory for Optimal Control Problems with Nondifferential Constraints J. Cloutier and C. D'Souza, Wright Lab., Eglin AFB, FL	AIAA-93-3750 Robust Neighboring External Guidance for the Advanced Launch System J. Bain and J. Speyer, Univ. of California, Los Angeles, CA	AIAA-93-3751 A Performance Comparison of Nonlinear Programming Algorithms for Large Sparse Problems J. Betts, S. Eldersveld, and W. Huffman, Boeing Computer Services, Seattle, WA	AIAA-93-3752 Matched Asymptotic Expansion of the Hamilton-Jacobi-Bellman Equation for Aeroassisted Plane-Change Maneuvers A. Calise and N. Mehlman, Georgia Institute of Technology, Atlanta, GA	AIAA-93-3753 An Investigation of the Fuel-Optimal Periodic Trajectories of a Hypersonic Vehicle L. Dewell and J. Speyer, Univ. of California, Los Angeles, CA		

Session 20-GNC-9—Spacecraft Controls I <i>Chaired by: G. SEVASTON, JPL, Pasadena, CA</i>							Bonsai II
AIAA-93-3754 Robust Attitude Control for Cassini Spacecraft Flying by Titan R. Chang, S. Lissman, E. Wong, P. Enright, W. Breckenridge, and M. Jahanshahi, JPL, Pasadena, CA	AIAA-93-3755 On Autonomous Imagery of Phobos and Deimos Using Flyby Camera Onboard Planet-B Spacecraft J. Kawaguchi, T. Hashimoto, I. Nakatani, and K. Ninomiya, Institute of Space & Astronautical Science, Kanagawa, Japan	AIAA-93-3756 Gyroless Fine Pointing on Small Explorer Spacecraft F. Bauer, NASA Goddard, Greenbelt, MD; W. Dellinger, Swales & Associates, Beltsville, MD	AIAA-93-3758 Attitude Dynamics of Spinning Rockets in Space with Internal Fluid Motion B. Staunton, Aerospace Corp., Los Angeles, CA; D. Mingori, Univ. of California, Los Angeles, CA	AIAA-93-3759 A Fuzzy-Based Attitude Controller for Flexible Spacecraft with On/Off Thrusters R. Knapp and N. Adams, Charles Stark Draper Lab., Cambridge, MA	AIAA-93-3760 Zero-Gyro Control of the International Ultraviolet Explorer J. O'Donnell Jr., NASA Goddard, Greenbelt, MD		

Monday Afternoon / August 9, 1993 (cont'd)

2:00	2:30	3:00	3:30	4:00	4:30	5:00	5:30
------	------	------	------	------	------	------	------

2:30	3:00	3:30	4:00	4:30	5:00	5:30
------	------	------	------	------	------	------

3:00	3:30	4:00	4:30	5:00	5:30
------	------	------	------	------	------

3:30	4:00	4:30	5:00	5:30
------	------	------	------	------

4:00	4:30	5:00	5:30
------	------	------	------

4:30	5:00	5:30
------	------	------

5:00	5:30
------	------

5:30

Session 21-GNC-10 — Integrated/Multidisciplinary Flight Control *Chaired by:* E. JACKSON, NASA Langley, Hampton, VA **Bonsai III**

Bonsai III

AIAA-93-3761 Integrated Control of Hypersonic Vehicles—A Necessity, Not Just a Possibility D. Schmidt, Arizona State Univ., Tempe, AZ	AIAA-93-3762 Robust Control of Hypersonic Vehicles Considering Propulsive and Aeroelastic Effects H. Buschek and A. Caise, Georgia Institute of Technology, Atlanta, GA	AIAA-93-3763 Dynamics of Hypersonic Flight Vehicles Exhibiting Significant Aeroelastic and Aeropropulsive Interactions F. Chavez and D. Schmidt, Arizona State Univ., Tempe, AZ	AIAA-93-3764 Identification of Integrated Airframe/Propulsion Effects for Application to Drag Minimization G. Schkolnik, NASA Dryden, Edwards, CA	AIAA-93-3765 Performance Seeking Control: Flight Program Overview and Future Directions G. Gagliard and J. Orme, NASA Dryden, Edwards, CA	AIAA-93-3766 Application of Controller Partitioning Optimization Procedure to Integrated Flight Propulsion Control Design for a STOVL Aircraft S. Garg, NASA Lewis, Cleveland, OH; P. Schmidt, Univ. of Akron, Akron, OH			
---	---	---	---	---	--	--	--	--

AIAA-93-3762 Robust Control of Hypersonic Vehicles Considering Propulsive and Aeroleastic Effects H. Buschek and A. Caïse, Georgia Institute of Technology, Atlanta, GA	AIAA-93-3763 Dynamics of Hypersonic Flight Vehicles Exhibiting Significant Aeroleastic and Aeropropulsive Interactions F. Chavez and D. Schmidt, Arizona State Univ., Tempe, AZ	AIAA-93-3764 Identification of Integrated Airframe/Propulsion Effects for Application to Drag Minimization G. Schkolnik, NASA Dryden, Edwards, CA	AIAA-93-3765 Performance Seeking Control: Flight Program Overview and Future Directions G. Gilyard and J. Orme, NASA Dryden, Edwards, CA	AIAA-93-3766 Application of Controller Partitioning Optimization Procedure to Integrated Flight Propulsion Control Design for a STOVL Aircraft S. Gang, NASA Lewis, Cleveland, OH; P. Schmidt, Univ. of Akron, Akron, OH	
---	---	---	--	--	--

<p>AIAA-93-3763 Dynamics of Hypersonic Flight Vehicles Exhibiting Significant Aeroleastic and Aeropropulsive Interactions F. Chavez and D. Schmidt. <i>Arizona State Univ., Tempe, AZ</i></p>	<p>AIAA-93-3764 Identification of Integrated Airframe/Propulsion Effects for Application to Drag Minimization G. Schkolnic, NASA Dryden, Edwards, CA</p>	<p>AIAA-93-3765 Performance Seeking Control: Flight Program Overview and Future Directions G. Gilyard and J. Orme, NASA Dryden, Edwards, CA</p>	<p>AIAA-93-3766 Application of Controller Partitioning Optimization Procedure to Integrated Flight Propulsion Control Design for a STOVL Aircraft S. Garg, NASA Lewis, Cleveland, OH; P. Schmidt, Univ. of Akron, Akron, OH</p>		
---	---	--	--	--	--

<p>AIAA-93-3764 Identification of Integrated Airframe/Propulsion Effects for Application to Drag Minimization G. Schkolnic, NASA Dryden, Edwards, CA</p>	<p>AIAA-93-3765 Performance Seeking Control: Flight Program Overview and Future Directions G. Gilyard and J. Orme, NASA Dryden, Edwards, CA</p>	<p>AIAA-93-3766 Application of Controller Partitioning Optimization Procedure to Integrated Flight Propulsion Control Design for a STOVL Aircraft S. Garg, NASA Lewis, Cleveland, OH; P. Schmidt, Univ. of Akron, Akron, OH</p>		
--	---	---	--	--

AIAA-93-3765 Performance Seeking Control: Flight Program Overview and Future Directions G. Gilyard and J. Orme. NASA Dryden, Edwards, CA	AIAA-93-3766 Application of Controller Partitioning Optimization Procedure to Integrated Flight Propulsion Control Design for a STOVL Aircraft S. Garg, NASA Lewis, Cleveland, OH; P. Schmidt, Univ. of Akron, Akron, OH		
--	--	--	--

<p>AIAA-93-3766 Application of Controller Partitioning Optimization Procedure to Integrated Flight Propulsion Control Design for a STOVL Aircraft S. Garg, NASA Lewis, Cleveland, OH; P. Schmidt, Univ. of Akron, Akron, OH.</p>			
--	--	--	--

Session 22-GNC-11 —Flexible Structure Modeling and Control Design *Chaired by:* R. LASKIN, JPL, Pasadena, CA **Redwood**

Redwood

<p>AIAA-93-3767 Hybrid Equations of Motion for Flexible Multibody Systems Using Quasicoordinates L. Merowitch and T. Stemple, Virginia Tech, Blacksburg, VA</p>	<p>AIAA-93-3768 Design of Stabilizing Control Laws for Mechanical Systems Based on Lyapunov's Method T. Ohtsuka and H. Fujii, Tokyo Metropolitan Institute of Technology, Tokyo, Japan</p>	<p>AIAA-93-3769 A Consistent Model Reduction of Modal Parameters for Reduced-Order Active Control K. Amini, K. Park, and L. Peterson, Univ. of Colorado, Boulder, CO</p>	<p>AIAA-93-3770 Dissipative Control of Energy Flow in Interconnected Systems Y. Kishimoto and D. Bernstein, Univ. of Michigan, Ann Arbor, M. S. Hall, MIT, Cambridge, MA</p>	<p>AIAA-93-3771 Optimal Slewing of Flexible Spacecraft with Prescribed Sensitivity of Terminal Structures J. Ben-Asher, IMI, Ramat Ha-Sharon, Israel</p>	<p>AIAA-93-3772 Modal Gramian Approach to Actuator and Sensor Placement for Flexible Structures K. Lim, NASA Langley, Hampton, VA, W. Sawrowski, JPL, Pasadena, CA</p>	<p>AIAA-93-3773 Robust Controller Design Based on Reduced-Order Models of Flexible Structures M. Gonzalez-Oberdoerfer and R. Craig Jr., Univ. of Texas, Austin, TX</p>
--	---	--	---	--	--	--

<p>AIAA-93-3768 Design of Stabilizing Control Laws for Mechanical Systems Based on Lyapunov's Method T. Ohtsuka and M. Fujii, <i>Tokyo Metropolitan Institute of Technology, Tokyo, Japan</i></p>	<p>AIAA-93-3769 A Consistent Modal Reduction of Modal Parameters for Reduced-Order Active Control J. Kim, K. Park, and L. Peterson, <i>Univ. of Colorado, Boulder, CO</i></p>	<p>AIAA-93-3770 Disipative Control of Energy Flow in Interconnected Systems Y. Kishimoto and D. Bernstein, <i>Univ. of Michigan, Ann Arbor, MI</i> K. S. Hui, MIT, Cambridge, MA</p>	<p>AIAA-93-3771 Optimal Slewing of Flexible Spacecraft with Prescribed Sensitivity of Terminal Control J. Ben-Asher, <i>IMI, Ramat Ha-Sharon, Israel</i></p>	<p>AIAA-93-3772 Model Gramman Approach to Actuator and Sensor Placement for Flexible Structures K. Lim, <i>NASA Langley, Hampton, VA</i> H. Sawronski, <i>JPL, Pasadena, CA</i></p>	<p>AIAA-93-3773 Robust Controller Design Based on Reduced-Order Models of Flexible Structures M. Gonzalez-Oberdorfer and R. Craig, Jr., <i>Univ. of Austin, TX</i></p>
---	---	---	--	--	--

<p>AIAA-93-3769 A Consistent Model Reduction of Modal Parameters for Reduced-Order Active Control A. Alvin, K. Park, and L. Peterson, Univ. of Colorado, Boulder, CO</p>	<p>AIAA-93-3770 Dissipative Control of Energy Flow in Interconnected Systems Y. Kishimoto and D. Bernstein, Univ. of Michigan, Ann Arbor, MI, S. Hall, MIT, Cambridge, MA</p>	<p>AIAA-93-3771 Optimal Slewing of Flexible Spacecraft with Prescribed Sensitivity of Terminal Control J. Ben-Asher, IMI, Ramat Ha-Sharon, Israel</p>	<p>AIAA-93-3772 Modal Gramman Approach to Actuator and Sensor Placement for Flexible Structures K. Lim, NASA Langley, Hampton, VA; M. Gavrinski, JPL, Pasadena, CA</p>	<p>AIAA-93-3773 Robust Controller Design Based on Reduced-Order Models of Flexible Structures M. Gonzalez-Oberdorferle and R. Craig Jr., Univ. of Texas, Austin, TX</p>
--	---	---	--	---

<p>AIAA-93-3770 Dissipative Control of Energy Flow in Interconnected Systems Y. Kishimoto and D. Bernstein, Univ. of Michigan, Ann Arbor, MI, St. Paul, MI, Cambridge, MA</p>	<p>AIAA-93-3771 Optimal Slewing of Flexible Spacecraft with Prescribed Sensitivity of Terminal Control J. Ben-Asher, IMI, Ramat Ha-Sharon, Israel</p>	<p>AIAA-93-3772 Modal Gramian Approach to Actuator and Sensor Placement for Flexible Structures K. Lim, NASA Langley, Hampton, VA; W. Gawronski, JPL, Pasadena, CA</p>	<p>AIAA-93-3773 Robust Controller Design Based on Reduced-Order Models of Flexible Structures M. González-Oberdörfer and R. Craig Jr., Univ. of Texas, Austin, TX</p>
--	---	--	---

<p>AIAA-93-3771 Optimal Slewing of Flexible Spacecraft with Prescribed Sensitivity of Terminal Control J. Ben-Asher, <i>IMI, Ramat Ha-Sharon, Israel</i></p>	<p>AIAA-93-3772 Modal Gramian Approach to Actuator and Sensor Placement for Flexible Structures K. Lim, <i>NASA Langley, Hampton, VA</i>, W. Gawronski, <i>JPL, Pasadena, CA</i></p>	<p>AIAA-93-3773 Robust Controller Design Based on Reduced-Order Models of Flexible Structures M. Gonzalez-Oberdiente and R. Craig Jr., <i>Univ. of Texas, Austin, TX</i></p>
--	--	--

AIAA-93-3772 Modal Gramian Approach to Actuator and Sensor Placement for Flexible Structures K. Lim, NASA Langley, Hampton, VA; W. Gawronski, JPL, Pasadena, CA	AIAA-93-3773 Robust Controller Design Based on Reduced-Order Models of Flexible Structures M. Gonzalez-Oberdierle and R. Craig Jr., Univ. of Texas, Austin, TX
---	--

AIAA-93-3773
Robust Controller Design
Based on Reduced-Order
Models of Flexible
Structures
M. Gonzalez-Oberdierle and
R. Craig Jr., *Univ. of Texas,*
Austin, TX

Session 23-GNC-12 —High-Performance Aircraft Control *Chaired by:* C-F. LIN, American GNC Corp., Chatsworth, CA **Ironwood**

Ironwood

<p>AIAA-93-3774 Robust, Nonlinear High Angle-of-Attack Control Design for a Supermaneuverable Vehicle J. Butington, R. Adams, and S. Bander, Wright Lab., Wright-Patterson AFB, OH</p>	<p>AIAA-93-3775 Cancellation Control Law for Lateral-Directional Dynamics of a Super Maneuverable Aircraft A. Snell, Univ. of California, Davis, CA</p>	<p>AIAA-93-3776 Design of a Flight Control System for a Highly Maneuverable Aircraft Using μ Synthesis G. Balas, J. Reiner, and W. Garrard, Univ. of Minnesota, Minneapolis, MN</p>	<p>AIAA-93-3777 Nonlinear Command Augmentation System for a High-Performance Aircraft P. Menon, <i>Optimal Synthesis</i>, Palo Alto, CA</p>	<p>AIAA-93-3778 Quasi-Optimal Steady State and Transient Maneuvers with and without Thrust Vectoring M. Dwyer, ANSER Corp., Arlington, VA; F. Lutz, Virginia Tech, Blacksburg, VA</p>	<p>AIAA-93-3779 Design of a Controller for a High-Performance Fighter Aircraft Using Robust Inverse Dynamic Estimation E. Muir, DRA Bedford, Bedfordshire, UK</p>
--	---	--	---	---	---

<p>AIAA-93-3775 Cancellation Control Law for Lateral-Directional Dynamics of a Supermaneuverable Aircraft A. Snell, <i>Univ. of California, Davis, CA</i></p>	<p>AIAA-93-3776 Design of a Flight Control System for a Highly Maneuverable Aircraft Using μ Synthesis G. Balas, J. Reiner, and W. Garrard, <i>Univ. of Minnesota, Minneapolis, MN</i></p>	<p>AIAA-93-3777 Nonlinear Command Augmentation System for a High-Performance Aircraft P. Menon, <i>Optimal Synthesis, Palo Alto, CA</i></p>	<p>AIAA-93-3778 Quasi-Optimal Steady State and Transient Maneuvers with and without Thrust Vectoring M. Dwyer, <i>ANSER Corp., Arlington, VA; F. Lutze, Virginia Tech, Blacksburg, VA</i></p>	<p>AIAA-93-3779 Design of a Controller for a High-Performance Fighter Aircraft Using Robust Inverse Dynamic Estimation E. Muir, <i>DRA Bedford, Bedfordshire, UK</i></p>	
---	---	---	---	--	--

<p>AIAA-93-3776 Design of a Flight Control System for a Highly Maneuverable Aircraft Using μ Synthesis G. Balas, J. Reiner, and W. Garrard <i>Univ. of Minnesota, Minneapolis, MN</i></p>	<p>AIAA-93-3777 Nonlinear Command Augmentation System for a High-Performance Aircraft P. Menon, <i>Optimal Synthesis, Palo Alto, CA</i></p>	<p>AIAA-93-3778 Quasi-Optimal Steady State and Transient Maneuvers with and without Thrust Vectoring M. Dwyer, <i>ANSER Corp., Arlington, VA; F. Lutze, Virginia Tech, Blacksburg, VA</i></p>	<p>AIAA-93-3779 Design of a Controller for a High-Performance Fighter Aircraft Using Robust Inverse Dynamic Estimation E. Muir, <i>DRA Bedford, Bedfordshire, UK</i></p>	
--	--	--	---	--

<p>AIAA-93-3777 Nonlinear Command Augmentation System for a High-Performance Aircraft P. Menon, <i>Optimal Synthesis</i>, Palo Alto, CA</p>	<p>AIAA-93-3778 Quasi-Optimal Steady State and Transient Maneuvers with and without Thrust Vectoring M. Dwyer, <i>ANSER Corp.</i>, Arlington, VA; F. Lutze, <i>Virginia Tech</i>, Blacksburg, VA</p>	<p>AIAA-93-3779 Design of a Controller for a High-Performance Fighter Aircraft Using Robust Inverse Dynamic Estimation E. Mur, <i>DRA Bedford</i>, Bedfordshire, UK</p>	
---	--	---	--

AIAA-93-3778 Quasi-Optimal Steady State and Transient Maneuvers with and without Thrust Vectoring M. Dwyer, ANSER Corp., Arlington, VA; F. Lutze, Virginia Tech, Blacksburg, VA	AIAA-93-3779 Design of a Controller for a High-Performance Fighter Aircraft Using Robust Inverse Dynamic Estimation E. Muir, DRA Bedford, Bedfordshire, UK		
--	--	--	--

<p>AIAA-93-3779 Design of a Controller for a High-Performance Fighter Aircraft Using Robust Inverse Dynamic Estimation E. Muir, DRA Bedford, Bedfordshire, UK</p>		
--	--	--

Session 24-GNC-13 —Missile Terminal Guidance *Chaired by:* S. BALAKRISHNAN, Univ. of Missouri, Rolla, MO **Cottonwood**

Cottonwood

<p>AIAA-93-3780 The Exact Closed-Form Solution of Generalized Proportional Navigation P.-J. Yuan, S.-C. Hsu, and C.-Y. Chang, <i>Chung Shan Institute of Science & Technology, Taiwan, ROC</i></p>	<p>AIAA-93-3781 When Bad Things Happen to Good Missiles P. Zarchan, <i>Charles Stark Draper Lab., Cambridge, MA</i></p>	<p>AIAA-93-3782 Guidance Law Based on an Exponential Criterion for Acceleration Constrained Missile and a Maneuvering Target I. Rusan, <i>Drexel Univ., Philadelphia, PA</i></p>	<p>AIAA-93-3783 Active Missile Guidance against Very-Low-Altitude Target T. Kuroda, F. Imado, and A. Ichikawa, <i>Mitsubishi Electric Corp., Amagasaki Hyogo, Japan</i></p>	<p>AIAA-93-3784 Optimal Guidance for a Short-Range Attack Missile D. Hull and D. Muzumdar, <i>Univ. of Texas, Austin, TX</i></p>	<p>AIAA-93-3785 Cut-Off Insensitive Guidance with Variable Time of Flight J. White, <i>Sandia National Labs., Albuquerque, NM</i></p>	<p>AIAA-93-3786 Analysis and Evaluation of Guidance Integrated Fuzing C. Yang and C.-F. Lin, <i>American GNC Corp., Chatsworth, CA</i> K. Chopper, H. Jaeger, L. Stephens, and D. Burdick, <i>Naval Air Warfare Center, China Lake, CA</i></p>
--	---	--	---	--	---	---

<p>AIAA-93-3781</p> <p>When Bad Things Happen to Good Missiles</p> <p>P. Zarchan, <i>Charles Stark Draper Lab., Cambridge, MA</i></p>	<p>AIAA-93-3782</p> <p>Guidance Law Based on an Exponential Criterion for Acceleration Constrained Missiles and a Maneuvering Target</p> <p>T. Ruskak, <i>Drexel Univ., Philadelphia, PA</i></p>	<p>AIAA-93-3783</p> <p>Active Missile Guidance against Very-Low-Altitude Target</p> <p>T. Kuroda, F. Imado, and A. Ichikawa, <i>Mitsubishi Electric Corp., Amagasaki Hyogo, Japan</i></p>	<p>AIAA-93-3784</p> <p>Optimal Guidance for a Short-Range Attack Missile</p> <p>D. Hull and D. Muzumdar, <i>University of Texas, Austin, TX</i></p>	<p>AIAA-93-3785</p> <p>Cut-Off Insensitive Guidance with Variable Time of Flight</p> <p>J. White, <i>Sandia National Labs., Albuquerque, NM</i></p>	<p>AIAA-93-3786</p> <p>Analysis and Evaluation of Guidance Integrated Fuzing</p> <p>C. Yang and C-F Lin, <i>American GNC Corp., Chatsworth, CA</i></p> <p>K. Choppen, H. Jaeger, L. Stephens, and D. Burdick, <i>Naval Air Warfare Center, China Lake, CA</i></p>
--	---	--	--	--	--

<p>AIAA-93-3782 Guidance Law Based on an Exponential Criterion for Acceleration Constrained Missiles and a Maneuvering Target</p> <p>I. Rusan, Drexel Univ., Philadelphia, PA</p>	<p>AIAA-93-3783 Active Missile Guidance against Very-Low-Altitude Target</p> <p>T. Kuroda, F. Imado, and A. Ichikawa, Mitsubishi Electric Corp., Amagasaki Hyogo, Japan</p>	<p>AIAA-93-3784 Optimal Guidance for a Short-Range Attack Missile</p> <p>D. Hull and D. Muzumdar, Univ. of Texas, Austin, TX</p>	<p>AIAA-93-3785 Cut-Off Insensitive Guidance with Variable Time of Flight</p> <p>J. White, Sandia National Labs., Albuquerque, NM</p>	<p>AIAA-93-3786 Analysis and Evaluation of Guidance Integrated Fuzing</p> <p>C. Yang and C-F. Lin, American GNC Corp, Chatsworth, CA; K. Chopper, H. Jaeger, L. Stephens, and D. Burdick, Naval Air Warfare Center, China Lake, CA</p>
--	--	---	--	---

AIIA-93-3783	AIIA-93-3784	AIIA-93-3785	AIIA-93-3786
Active Missile Guidance	Optimal Guidance for a	Cut-Off Insensitive Guidance	Analysis and Evaluation of
Against Very-Low-Altitude	Short-Range Attack Missile	With Variable Time of Flight	Guidance Integrated Fuzing
Target	D. Hull and D. Muzumdar,	J. White, Sandia National	C. Yang and C.-F. Lin, American
T. Kuroda, F. Imado, and	Univ. of Texas, Austin, TX	Labs., Albuquerque, NM	Ginger Corp., Chatsworth, CA.
A. Ichikawa, Mitsubishi Electric			K. Chopper, H. Jaser,
Corp., Amagasaki Hyogo,			L. Stephens, and D. Burdick,
Japan			Naval Air Warfare Center,
			China Lake, CA

<p>AIAA-93-3784 Optimal Guidance for a Short-Range Attack Missile D. Hull and D. Muzumdar, Univ. of Texas, Austin, TX</p>	<p>AIAA-93-3785 Cut-Off Insensitive Guidance with Variable Time of Flight J. White, Sandia National Labs., Albuquerque, NM</p>	<p>AIAA-93-3786 Analysis and Evaluation of Guidance Integrated Guidance C. Yang and C-F Lin, American GNC Corp, Chatsworth, CA W. Chopper, H. Jaeger, L. Stephens, and D. Burdick, Naval Air Warfare Center, China Lake, CA</p>	
--	--	--	--

<p>AIAA-93-3785 Cut-Off Insensitive Guidance with Variable Time of Flight J. White, Sandia National Labs., Albuquerque, NM</p>	<p>AIAA-93-3786 Analysis and Evaluation of Guidance Integrated Fuzing C. Yang and C-F. Lin, American GNC Corp, Chatsworth, CA; K. Chopper, H. Jaeger, L. Stephens, and D. Burdick, Naval Air Warfare Center, China Lake, CA</p>
--	--

<p>AIAA-93-3786 Analysis and Evaluation of Guidance Integrated Fuzing C. Yang and C-F. Lin, American GNC Corp, Chatsworth, CA; K. Chopper, H. Jaeger, L. Stephens, and D. Burdick, Naval Air Warfare Center, China Lake, CA</p>	
--	--

Session 25-GNC-14 —Challenges to GN&C Education for Next Decade *Chaired by: J. VAGNERS, Univ. of Washington, Seattle, WA* **Colton III**

Colton III

Papers and authors to be announced.

Session 26-AFM-5—Wind Shear and Atmospheric Disturbances <i>Chaired by: F. GARRETT JR., Arizona State Univ., Tempe, AZ</i>						Ferrante I
AIAA-93-3641 Wind-Tunnel Investigation of Wind-Shear Effect on Turning Flight M. Ghazi, King Abdulaziz Univ., Jeddah, Saudi Arabia	AIAA-93-3642 A Simulation Study of the Interaction between a Wake Vortex and an Encountering Airplane E. Stewart, NASA Langley, Hampton, VA	AIAA-93-3643 A Windshear Hazard Window Useful in Studying the Effects of Wind Shear on the Airplane During the Landing Approach R. White, Bellevue, WA	AIAA-93-3644 Effect of Rotary Atmospheric Gusts on Fighter Airplane M. Ghazi, King Abdulaziz Univ., Jeddah, Saudi Arabia			
Session 27-AFM-6—Aircraft Handling Qualities I <i>Chaired by: R. MEYER, Lockheed Corp., Marietta, GA</i>						Ferrante III
AIAA-93-3645 Development of Flying Qualities and Agility Evaluation Maneuvers D. Wilson, D. Riley, and K. Citurs, McDonnell Douglas Aerospace, St. Louis, MO, T. Cord, Wright Lab, Wright-Patterson AFB, OH	AIAA-93-3646 The Evaluation of Several Agility Metrics for Fighter Aircraft Using Optimal Trajectory Analysis G. Ryan III, PRC Space, Edward, CA, D. Downing, Univ. of Kansas, Lawrence, KS	AIAA-93-3647 NASA/NAVY Investigation of Lateral-Directional Control Requirements for High-Performance Fighter/Attack Aircraft J. Foster, H. Ross, and P. Ashley, NASA Langley, Hampton, VA	AIAA-93-3648 Aircraft Control Requirements and Achievable Dynamics Prediction J. Boland, D. Riley, and K. Citurs, McDonnell Douglas Aerospace, St. Louis, MO	AIAA-93-3649 Investigation of Roll Requirements for Carrier Approach K. Citurs, K. Doll, and J. Buckley, McDonnell Douglas Aerospace, St. Louis, MO	AIAA-93-3650 Lateral-Directional Departure Criteria F. Lutz, W. Durham, and W. Mason, Virginia Tech, Blacksburg, VA	
Session 28-AFM-7—Projectile/Missile Aerodynamics: Experiments and Analysis <i>Chaired by: J. GRAU, U.S. Army ARDEC, Picatinny Arsenal, NJ</i>						Ferrante II
AIAA-93-3651 Ring Wing for an Underwater Missile H. August, Hughes Missile Systems Co., Canoga Park, CA, E. Carapezza, Arlington, VA	AIAA-93-3652 Segmented Penetrator Flight Dynamics E. Schmidt, U.S. Army Research Lab., Aberdeen Proving Ground, MD	AIAA-93-3653 The Aerodynamic Characteristics of Rhombic Cross-Sectioned Bodies with and without Fins G. Maliniak and A. Sigal, Technion-IIT, Haifa, Israel	AIAA-93-3654 Free-Spin Damping Measurement Techniques E. Marquart, Aerospace Systems Facility, Arnold AFB, TN	AIAA-93-3655 A Hybrid Method for the Analysis of Multiple-Fin Configurations A. Sigal, Technion-IIT, Haifa, Israel	AIAA-93-3656 Ring-Wing Missile for Compressed Carriage on an Aircraft H. August, Hughes Missile Systems Co., Canoga Park, CA, R. Osborn and M. Pinney, Wright Lab., Wright-Patterson AFB, OH	
Session 29-AFM-8—Workshop on Trajectory Optimization Methods and Applications <i>Chaired by: K. LANGAN, Wright Lab., Wright-Patterson AFB, OH</i>						Colton I & II
This workshop is intended to present recent advances in trajectory optimization programming and applications. The developers of the Optimal Trajectories by Implicit Simulation (OTIS) and Program to Optimize Simulated Trajectories (POST) have been invited to present recent improvements. Other trajectory optimization code developers, as well as users of trajectory optimization programs, are also welcome to present any advancements, modifications, and/or applications. The workshop format will consist of short 15–20 minute informal presentations, followed by an open discussion.						
Session 30-FST-4—Human Factors and Pilot Cueing <i>Chaired by: F. CARDULLO, SUNY, Binghamton, NY</i>						Los Angeles
AIAA-93-3560 Texture as a Visual Cueing Element in Computer Image Generation. Part I—Representation of the Sea Surface J. Sinakun, John B. Smacori Associates, Pebble Beach, CA	AIAA-93-3561 What Optical Cues Do Pilots Use to Initiate the Landing Flare: Results of a Piloted Simulator Experiment J. Van der Vaart and S. Advani, Delft Univ. of Technology, Delft, The Netherlands	AIAA-93-3562 Acoustical and Vibratory Stimuli Interdependencies and Their Applications in Simulation and Cue Synchronization R. Sawler and R. Matusof, CAE-Link Corp., Binghamton, NY	AIAA-93-3563 Transport Delay Compensation: An Inexpensive Alternative to Increasing Image Generator Update Rate F. Cardullo and G. George, SUNY, Binghamton, NY	AIAA-93-3564 Multitask Integration Model for Human Ergonomics Perception G. Zacharias and A. Miao, Charles River Analytics, Cambridge, MA	AIAA-93-3565 Technique Problems of Visual and Motion Cueing Simulation V. Rodchenko and A. Predtechensky, TsAGI, Moscow, Russia	

Monday Afternoon / August 9, 1993 (cont'd)							
2:00	2:30	3:00	3:30	4:00	4:30	5:00	5:30
Session 31-FST-5—Modeling the Simulation Environment <i>Chaired by: S. ADVANI, Delft Univ. of Technology, Delft, The Netherlands</i>							Santa Monica
AIAA-93-3566 Visual Weather Simulation Using Meteorological Databases B. Montag, Southwest Research Institute, San Antonio, TX	AIAA-93-3567 Line of Sight Determination in Real-Time Simulations F. Kull and D. Foughti, SYRE, Moffett Field, CA	AIAA-93-3568 The Challenges of Simulating Wake Vortex Encounters and Assessing Separation Criteria E. Dunham, R. Sluever, and D. Victory, NASA Langley, Hampton, VA	AIAA-93-3607 Terrain Modeling for Real-Time Photo Texture Based Visual Simulation V. Devarajan and D. McArthur, Univ. of Texas, Arlington, TX				
Session 32-FST-6—Simulation Applications <i>Chaired by: D. KEY, U.S. Army Aviation Systems Command, Moffett Field, CA</i>							San Diego
AIAA-93-3569 Fly-by-Throttle Transport Simulation Controller Comparison Study J. Conley, NASA Ames, Moffett Field, CA	AIAA-93-3570 A Primary Flight Display for Four-Dimensional Guidance and Navigation E. Theunissen, Delft Univ. of Technology, Delft, The Netherlands	AIAA-93-3571 The Use of a Simulator in Aircraft Accident Investigation T. Nieuwpoort and J. van Hengst, Fokker Aircraft B.V., Amsterdam, The Netherlands	AIAA-93-3608 A Comparative Evaluation of Three Take-Off Performance Monitor Display Types J. Vesspay and R. Khata, NRL, Amsterdam, The Netherlands	AIAA-93-3572 Prognostic Evaluation of Getting Trauma Induced by Impact Accelerations U. Mazurin and G. Stypakov, Institute of Aviation & Space Medicine, Moscow, Russia			
Session 33-AA-5—Unsteady Aerodynamics <i>Chaired by: J. ASHWORTH, Pentagon, Washington, DC</i>							San Carlos I
AIAA-93-3422 Separated Flowfield and Lift on an Airfoil with an Oscillating Leading-Edge Flap K. Laurie and S. Farokhi, KU Center for Research, Lawrence, KS	AIAA-93-3423 Unsteady Ground Effects on Aerodynamic Coefficients of Finite Wings with Camber A. Nuhait, King Saud Univ., Riyadh, Saudi Arabia	AIAA-93-3424 The Moving Wall Effect vis-a-vis Other Dynamic Stall Flow Mechanisms L. Ericsson, Mt. View, CA	AIAA-93-3425 A Visualization Study of the Vortical Flow over a Double-Delta Wing in Dynamic Motion S. Hebbbar and M. Platzer, Naval Postgraduate School, Monterey, CA; F. H. Li, Republic of China Navy, Taiwan, ROC	AIAA-93-3426 Neural Network Prediction of Three-Dimensional Unsteady Separated Flow Fields S. Schreck, USAF Academy, Colorado Springs, CO; W. Faller, Bioscience Space Technologies, Boulder, CO; M. Lutiges, Dept of Aerospace Engineering Sciences, Boulder, CO	AIAA-93-3427 Effect of Canard Oscillations on the Vortical Flowfield of an X-31A-Like Fighter Model in Dynamic Motion S. Hebbbar, M. Platzer, and D.-M. Liu, Naval Postgraduate School, Monterey, CA		
Session 34-AA-6—CFD/Grid Aerodynamics <i>Chaired by: S. REZNICK, Systems Engineering, Wright-Patterson AFB, OH</i>							San Carlos II
AIAA-93-3428 Overlapping Grid Methods for High-Lift and Store Carriage Application M. Mani and R. Bush, McDonnell Douglas Aerospace, St. Louis, MO	AIAA-93-3429 Transitioning Full Navier-Stokes Predictions of Sling and Distortion Increments for a Fighter Aircraft into a Production Environment R. Dyer, G. Fintrock, and B. Robinson, McDonnell Douglas Aerospace, St. Louis, MO	AIAA-93-3430 New-Field Supersonic Flow Predictions by an Adaptive Unstructured Tetrahedral Grid Solver J. Djomehri and L. Erickson, NASA Ames, Moffett Field, CA	AIAA-93-3431 A Semi-Implicit Finite Volume Scheme for Solution of Euler Equations on Three-Dimensional Prismatic Grids S. Pandya and M. Hafez, Univ. of California, Davis, CA	AIAA-93-3432 Calculations on Double-Fin Interactions at High Speed D. Gatonde and J. Shang, Wright Lab., Wright-Patterson, AFB, OH	AIAA-93-3433 A Zonal CFD Method for Three-Dimensional Wing Simulations M. Summa, D. Strash, and S. Yoo, Analytical Methods, Redmond, WA		

Session 35-AA-7 —Experimental Aerodynamics <i>Chaired by: G. WINCHENBACH, Wright Lab., Eglin AFB, FL</i>						San Carlos III	
AIAA-93-3434 Flow Field Measurements in a Crossing Shockwave Turbulent Boundary Layer Intersection at Mach 3 J. Lachowicz and N. Chokani, North Carolina State Univ., Raleigh, NC	AIAA-93-3435 Scaling of Incipient Separation in High-Speed Laminar Flows G. Inger, Cranfield Institute of Technology, Cranfield, UK	AIAA-93-3436 The Effect of a High-Thrust Pusher Propeller on the Flow over a Straight Wing F. Catalano and J. Sjolbery, Cranfield Institute of Technology, Cranfield, UK	AIAA-93-3437 Analysis of the Effect of Surface Heating on Boundary Layer Development over a NASA NLS-G215 Airfoil S. Schmid and B. Selberg, Univ. of Missouri, Rolla, MO	AIAA-93-3438 Oscillating Trailing Edge Controls on Delta Wings for Various Incidence D. Pilkington and N. Wood, Univ. of Bath, Bath, UK	AIAA-93-3439 Application of a Two-Camera Video Imaging System to Three-Dimensional Vortex Tracking in the 80-by-120-Foot Wind Tunnel L. Meyn and M. Bennett, NASA Ames, Moffett Field, CA		
Session 36-AA-8 —Hypersonic Aerodynamics <i>Chaired by: J. ZIEN, Naval Surface Warfare Center, Silver Spring, MD</i>						San Carlos IV	
AIAA-93-3440 A Visualizing Method of Streamline around Hypersonic Vehicles N. Nishio, Fukuyama Univ., Fukuyama, Japan	AIAA-93-3441 Rarefied-Flow Shuttle Aerodynamics Flight Model R. Blanchard, NASA Langley, Hampton, VA; K. Larman and C. Moats, Lockheed Engineering & Sciences Co., Hampton, VA	AIAA-93-3442 Waveriders with Finlets X. He, M. Rasmussen, and R. Cox, Univ. of Oklahoma, Norman, OK	AIAA-93-3443 Hypersonic Aerodynamic Characteristics for Langley Test Technique Demonstrator P. Phillips and C. Cruz, NASA Langley, Hampton, VA	AIAA-93-3444 Aerodynamic Design of a Hypersonic Body with a Constant Favorable Pressure Gradient F. Shope and R. Spinetti, Calspan Corp., Arnold AFB, TN	AIAA-93-3445 Hypersonic Viscous Aerodynamics Using Improved Bridging Procedures E. Knox and A. Jain, Remtech, Huntsville, AL; C. Seaford, NASA Marshall, Huntsville, AL		
5:00 – 6:00 pm Plenary — Flight Simulation in China YOU CHONGLIN, Beijing Aviation Simulator Co.						Steinbeck Forum	

Tuesday Morning / August 10, 1993							
9:00	9:30	10:00	10:30	11:00	11:30	12:00	12:30

8:00 am — 9:00 am Plenary — Guidance, Navigation and Control — From Instrumentation to Information Management ELI GAI, Charles Stark Draper Laboratory, Cambridge, MA	Steinbeck Forum
---	-----------------

Session 37-GNC-15—Spacecraft Guidance and Navigation I <i>Chaired by: P. RAO, Martin Marietta Corp., Denver, CO</i>						Bonsai I	
AIAA-93-3787 Performance Testing of a GPS-Based Attitude Determination System P. Axelrad and B. Chesley, Univ. of Colorado, Boulder, CO	AIAA-93-3788 Application of GPS Attitude Determination for Gravity Gradient Stabilized Spacecraft G. Lightsey, C. Cohen, and B. Parkinson, Stanford Univ., Stanford, CA	AIAA-93-3789 Solutions for the Geostationary Collocation Problem U. Bruege and T. Goriach, Deutsche Aerospace, Bremen, Germany; P. Taylor, Vega Space Systems, UK	AIAA-93-3790 Reentry Control to a Drag vs Energy Profile A. Roemerke and A. Markl, Univ. of Stuttgart, Stuttgart, Germany	AIAA-93-3791 Self-Tuning Guidance Applied to Aeroassisted Plane Change Problems S. Kamarsu and S. Balakrishnan, Univ. of Missouri, Rolla, MO			
Session 38-GNC-16—Computational Control and Dynamics II <i>Chaired by: E. CHEN, Charles Stark Draper Lab., Cambridge, MA</i>						Bonsai II	
AIAA-93-3794 On the Use of the Constant Mass Matrix in Flexible Multibody Systems T. Anthony, NASA Goddard, Greenbelt, MD; R. Craig Jr., Univ. of TX, Austin, TX	AIAA-93-3795 Order Reduction of Aeroelastic Models through LK Transformation and Riccati Iteration L. Anderson, Boeing Commercial Airplane Group, Seattle, WA	AIAA-93-3796 Modal Weighting of Linear Quadratic Regulators D. Mackison and E. Wall, Univ. of Colorado, Boulder, CO					

Tuesday Morning / August 10, 1993 (cont'd)							
9:00	9:30	10:00	10:30	11:00	11:30	12:00	12:30
Session 39-GNC-17—GNC Components and Avionics <i>Chaired by: D. BATES JR., Charles Stark Draper Lab., Cambridge, MA</i>							Bonsai III
AIAA-93-3797 Comparison of Classical and H _∞ Controller Design for a Single-Axis Prototype Electromechanical Actuator G. Hartley, Naval Air Warfare Center, China Lake, CA	AIAA-93-3798 The LN200 Fiber Gyro Based Tactical Grade IMU G. Pavlath, Litton Guidance & Control Systems, Woodland Hills, CA	AIAA-93-3799 Micromechanical Inertial Measurement Units (MMIMU) J. Sitomer, J. Connelly, and J. Martin, Charles Stark Draper Lab., Cambridge, MA	AIAA-93-3800 Vibrating Beam Accelerometer Performance—Strategic and High-g Applications R. Cimeria and R. Weber, Kearliff Guidance & Navigation Corp., Wayne, NJ	AIAA-93-3801 Dynamic Attitude Measurement System L. Wilk and T. Hamilton, Charles Stark Draper Lab., Cambridge, MA			
Session 40-GNC-18—Robust Control <i>Chaired by: S. HALL, Univ. of Michigan, Ann Arbor, MI</i>							Redwood
AIAA-93-3802 Design of Low Sensitivity and Norm Multivariable Controller Using Eigenstructure Assignment and the Method of Inequalities R. Patton, G. Liu, and J. Chen, Univ. of York, York, UK	AIAA-93-3803 Robust Stabilization of the Space Station in the Presence of Inertia Matrix Uncertainty B. Wie and O. Liu, Arizona State Univ., Tempe, AZ; J. Sunkei, NASA Johnson, Houston, TX	AIAA-93-3804 Robust Fuel- and Time-Optimal Control of Uncertain Flexible Space Structures B. Wie and R. Sinha, Arizona State Univ., Tempe, AZ; J. Sunkei and K. Cox, NASA Johnson, Houston, TX	AIAA-93-3805 Robustified Control Design by an Approximated Min-Max Technique M. Fischer and M. Psiaki, Cornell Univ., Ithaca, NY	AIAA-93-3806 A Robust Controller for Second-Order Systems Using Acceleration Measurements C. Chuang and O. Courouge, Georgia Institute of Technology, Atlanta, GA; J.-N. Juang, NASA Langley, Hampton, VA	AIAA-93-3807 Implementation of the Eigensystem Realization Algorithm without the Formation of the Hankel Matrix or Its Square R. Quan, Frank H. Seiler Lab., USAF Academy, CO		
Session 41-GNC-19—Flight/Propulsion Control Integration and Issues <i>Chaired by: W. MERRILL, NASA Lewis, Cleveland, OH</i>							Ironwood
AIAA-93-3808 Integrated Flight/Propulsion Control: Subsystem Specifications for Performance K. Neighbors and S. Rock, Stanford Univ., Stanford, CA	AIAA-93-3809 A Comparative Study of Multivariable Robustness Techniques as Applied to Integrated Flight and Propulsion Control J. Schierman, A. Lovell, and D. Schmidt, Arizona State Univ., Tempe, AZ	AIAA-93-3810 TVC Control for the AIAA Design Challenge Airplane M. Dierks, Southern Illinois Univ., Edwardsville, IL; K. Wise, McDonnell Douglas Aerospace, St. Louis, MO	AIAA-93-3811 Antiwindup Analysis and Design Approaches for MIMO Systems V. Marcopoli and S. Phillips, Case Western Reserve Univ., Cleveland, OH	AIAA-93-3812 A Linear, Multivariable, Limit and Integrator Wind-Up Protection Design Method D. Mattern, Sverdrup Technology, Cleveland, OH	Flight Test of a Fly By Throttle Flight Control System on an F-15 NASA Airplane (Invited Paper) G. Gilyard and F. Burcham, NASA Dryden Flight Research Facility, Edwards, CA		
Session 42-GNC-20—Flight Control Design and Analysis <i>Chaired by: V. CHENG, NASA Ames, Moffett Field, CA</i>							Cottonwood
AIAA-93-3814 A Pseudo-Loop Design Strategy for the Longitudinal Control of Hypersonic Aircraft P. Vu and D. Biezad, California Polytechnic State Univ., San Luis Obispo, CA	AIAA-93-3815 Longitudinal and Lateral-Directional Flying Qualities Investigation of High-Order Characteristics for Advanced Technology Transports K. Rossitto and J. Hodgkinson, McDonnell Douglas Aerospace, Long Beach, CA	AIAA-93-3816 Initial Results of an In-Flight Investigation of Longitudinal Flying Qualities for Augmented Transport S. Rock, H. Ashley, R. Digumarthy, and K. Chaney, Stanford Univ., Stanford, CA	AIAA-93-3817 Active Control for Fin Buffet Alleviation S. Rock, H. Ashley, R. Digumarthy, and K. Chaney, Stanford Univ., Stanford, CA	AIAA-93-3818 Guidance and Control Law Design for Automatic Landing Flight Experiment of Reentry Space Vehicle Y. Miyazawa, K. Ishikawa, and K. Fujii, National Aerospace Lab., Tokyo, Japan			

Session 43-AFM-9—Aircraft Trajectory Optimization <i>Chaired by: K. Bilimoria, Arizona State Univ., Tempe, AZ</i>						Ferrante I
AIAA-93-3657 Nonsmooth Trajectory Optimization: An Approach Using Continuous Simulated Annealing P. Lu and A. Khan, Univ. of Iowa, Ames IA	AIAA-93-3658 Optimal Turns with Altitude Variations B. Järnmark, Saab Military Aircraft, Linköping, Sweden	AIAA-93-3659 Optimum Poststall Turning and Supersonic Turning N. Vinh and Y. Tseng, Univ. of Michigan, Ann Arbor, MI	AIAA-93-3660 Aeropropulsive System Optimization in Hypersonic Flight D. Schmidt, K. Bilimoria, and A. Lovell, Arizona State Univ., Tempe, AZ			
Session 44-AFM-10—Aircraft Dynamics and Performance I <i>Chaired by: U. SHANKAR, RCA Satellite Systems, Princeton, NJ</i>						Ferrante II
AIAA-93-3661 Results and Lessons Learned from Two Wright Laboratory Flight Research Programs (Invited-1 hour) D. Moorhouse and L. Walchli, Wright Lab., Wright-Patterson AFB, OH	AIAA-93-3662 A Simplified Wing Rock Prediction Method B. Liebst, AFIT, Wright-Patterson AFB, OH; R. Nolan, USAF, Edwards AFB, CA	AIAA-93-3663 Dynamic Modeling of a Trailing-Wire Cable Towed by an Orbiting Aircraft L. Schmidt, Naval Postgraduate School, Monterey, CA; J. Clifton, USN, Lexington Park, MD; T. Stuart, USN, Arlington, VA	AIAA-93-3664 Aeroelastic Effects on the B-2 Maneuver Response B. Winther, D. Hagemeyer, R. Britt, and W. Rodden, Northrop Corp., Pico Rivera, CA	AIAA-93-3665 A Feedforward Logic for the I/O Map of Aircraft Inverse Problems G. Mattioli, Univ. of Pisa, Pisa, Italy; L. de Socio and A. Leonessa, Univ. of Rome La Sapienza, Rome, Italy		
Session 45-AFM-11—Aircraft Handling Qualities II <i>Chaired by: J. FOSTER, NASA Langley, Hampton, VA</i>						Ferrante III
AIAA-93-3666 Rotocraft Handling Qualities in Turbulence R. Hess, Univ. of California, Davis, CA	AIAA-93-3667 A Method of Wind Shear Detection for Powered-Lift STOL Aircraft K. Funabiki, T. Bando, and K. Tanaka, National Aerospace Lab., Tokyo, Japan; C. Hynes and G. Hardy, NASA Ames, Moffett Field, CA	AIAA-93-3668 A New Flying Qualities Criterion for Flying Wings W. Monnich, DLR, Braunschweig, Germany	AIAA-93-3669 Pilots' Control Behavior Including Feedback Structures Identified by an Improved Method N. Goto, Kyushu Univ., Fukuoka, Japan	AIAA-93-3670 Wind-Shear Endurance Capability for Powered-Lift STOL Aircraft T. Bando and K. Tanaka, National Aerospace Lab., Tokyo, Japan; C. Hynes and G. Hardy, NASA Ames, Moffett Field, CA	AIAA-93-3671 F/A-18 Departure Recovery Improvement Evaluation, J. Lackey and M. Stevens, Naval Air Warfare Center, Patuxent River, MD	
Session 46-AFM-12—Workshop on Atmospheric Flight Mechanics Education <i>Chaired by: R. NELSON, Univ. of Notre Dame, Notre Dame, IN</i>						Colton I & II
<p>This workshop provides an opportunity for informal discussion of all aspects of atmospheric flight mechanics (AFM) education at both undergraduate and graduate levels in aerospace engineering curricula. Suggested topics that may be addressed include the following: Are the traditional AFM courses in aerodynamics, stability and control, and aerospace vehicle design still relevant and adequate in meeting the present industry needs? What changes or additions would the industry like to see? What kinds of support would the universities like to receive from industry or government agencies to advance the state-of-the-art in AFM education? How can AFM education keep up with the fast pace of research and advances in this area, especially in designing an undergraduate curriculum? Discussion of innovative and realistic approaches to AFM education with emphasis on practical and design experience of the graduating student are of particular interest. The workshop format consists of short (10 minutes) presentations, followed by open discussions.</p>						
Session 47-FST-7—Simulation Facilities 2 <i>Chaired by: L. KNOTTS, Calspan Corp., Buffalo, NY</i>						Santa Monica
AIAA-93-3573 The Advanced Concepts Flight Simulator: A Full-Mission Simulation Research Tool for Human Factors B. Sullivan, R. Shiner, and R. Pando, NASA Ames, Moffett Field, CA	AIAA-93-3574 The Development of a Simulator Facility for Advanced Research Into Simulation Techniques, Motion System Control and Navigation Systems Technologies S. Advani, Delft Univ. of Technology, Delft, The Netherlands	AIAA-93-3575 Development and Operation of a Real-Time Simulation at the NASA Ames Vertical Motion Simulator C. Sweeney, S. Sheppard, and M. Chetelat, SYRE, Moffett Field, CA	AIAA-93-3576 An Asynchronous Simulation Methodology for Improving the Performance of Training Simulators R. Howe, Univ. of Michigan, Ann Arbor, MI; E. Fadden and W. Deiss, Applied Dynamics International, Ann Arbor, MI	AIAA-93-3577 Semi-Full-Scale Dynamic Simulation Complex on the Basis of Centrifuge O. Yakimenko and V. Vasiletz, Zhukovskiy Military Air Force, Moscow, Russia		

Tuesday Morning / August 10, 1993 (cont'd)

9:00	9:30	10:00	10:30	11:00	11:30	12:00	12:30
------	------	-------	-------	-------	-------	-------	-------

Session 48-FST-6—Motion Systems *Chaired by: J. SINACORI, John B. Sinacori Associates, Pebble Beach, CA*

Los Angeles

AIAA-93-3578 False Cue Detection Thresholds in Flight Simulation R. Hosman and H. van der Steen, <i>Delft Univ. of Technology, Delft, The Netherlands</i>	AIAA-93-3579 Simulation Motion Effects on Single Axis Compensatory Tracking J. Schroeder, <i>NASA Ames, Moffett Field, CA</i>	AIAA-93-3580 Pilot Evaluations of Augmented Flight Simulator Motion S. Garrood and L. Reid, <i>Univ. of Toronto, Downsview, Ontario, Canada</i>	AIAA-93-3581 New Concept of the Motion System for the Low-Cost Flight Simulator Development and Design V. Shibaev, <i>TsAGI, Moscow Region, Russia</i>				
---	---	---	--	--	--	--	--

Session 49-FST-9—Simulation Networking *Chaired by: B. GOLDIEZ, Institute for Simulation & Training, Orlando, FL*

San Diego

AIAA-93-3582 Representation of Vehicle Location in Networked Simulations K.-C. Lin, H. Ng, and B. Goldiez, <i>Institute for Simulation & Training, Orlando, FL</i>	AIAA-93-3583 Implementation of Expert Systems within an Interactive Tactical Environment D. Siksik, <i>CAE Electronics, Montreal, Quebec, Canada</i>	AIAA-93-3584 Dissimilar Computer Image Generators Provide Challenge in Development of Apache Longbow Full-Mission Engineering Simulator J. Richard, <i>McDonnell Douglas Training Systems, Mesa, AZ</i>	AIAA-93-3585 Pseudo Aircraft Systems: A Multi-Aircraft System for Air Traffic Control Research R. Weske and G. Danek, <i>NASA Ames, Moffett Field, CA</i>				
--	--	---	---	--	--	--	--

Session 50-AA-9—Wing/Forebody/Vortex/Aerodynamics *Chaired by: H. HOEIJMAKERS, NLR, Amsterdam, The Netherlands*

San Carlos I

AIAA-93-3446 Aerodynamic Characteristics of a Delta Wing with a Body-Hinged Leading-Edge Extension E. Sytnerud, S. Farokhi, and R. Toghiani, <i>Univ. of Kansas, Lawrence, KS; D. Neuhaert, Lockheed Engineering & Science Co., Hampton, VA</i>	AIAA-93-3447 On the Aerodynamics and Performance of Active Vortex Generators R. Barrett and S. Farokhi, <i>Univ. of Kansas, Lawrence, KS</i>	AIAA-93-3448 An Experimental Investigation of Tip Vortices at Mach 2.5 F. Wang and P. Storza, <i>Polytechnic Univ., Farmingdale, NY</i>	AIAA-93-3449 Forebody Vortex Control with Jet and Slot Blowing on an F/A-18 B. Kramer, C. Suarez, and G. Malcolm, <i>Edelics International, Torrance, CA; K. Jones, Sterling Federal Systems, Inc., Palo Alto, CA</i>	AIAA-93-3450 Forebody Vortex Control on a F/A-18 Using Small, Rotatable "Tip-Strakes" C. Suarez, B. Kramer, and G. Malcolm, <i>Edelics International, Torrance, CA</i>	AIAA-93-3451 Prediction of Forebody Vortex-Induced Wing Rock on Wing-Body Configurations M. Mendenhall, S. Perkins Jr., and L. Ericsson, <i>Nielsen Engineering & Research, Mountain View, CA</i>		
---	--	---	---	--	---	--	--

Session 51-AA-10—CFD Grid Methods *Chaired by: D. SMYTH, McDonnell Aircraft Co., Long Beach, CA*

San Carlos II

Aspects of Numerical Grid Generation: Current Science and Art (Invited-1 hr) J. Thompson, <i>Mississippi State Univ., Starkville, MS; N. Weatherill, Univ. of Swansea, Wales, UK</i>	AIAA-93-3452 Unstructured Grid Generation Using Interactive Three-Dimensional Boundary and Efficient Three-Dimensional Volume Methods T. Gatzke, <i>McDonnell Douglas Aerospace, St. Louis, MO; N. Weatherill, Univ. College of Swansea, Wales, UK</i>	AIAA-93-3453 Unstructured Viscous Grid Generation by Advancing—Layers Method S. Pirzadeh, <i>Vigyan, Hampton, VA</i>	AIAA-93-3454 Unstructured Grid on Nubus Surfaces J. Samareh-Abolhasani, <i>Computer Sciences Corp., Hampton, VA</i>	AIAA-93-3455 Computation of Viscous Flows Using UNIVERSE-Series, Unstructured Grid Flow Solver S. Chakravarthy, K.-Y. Szema, S. Ramakrishnan, and U. Goldberg, <i>Rockwell International Science Center, Thousand Oaks, CA</i>		
--	--	--	---	--	--	--

Session 52-AA-11 —Missile Aerodynamics						Chaired by: M. MILLER, Research Directorate, Aberdeen Proving Ground, MD		San Carlos III			
AIAA-93-3456 Wind-Tunnel Tests of an Inclined Cylinder Having Helical Grooves T. Stuart, USN, Springfield, VA; J. Clifton, USN, Lexington Park, MD; L. Schmidt, Naval Postgraduate School, Monterey, CA	AIAA-93-3457 Free Spin Damping Measurement Techniques E. Marquardt, Calspan Corp., Arnold AFB, TN	AIAA-93-3458 Air Data Estimation System for Control of Guided Munitions M. Anderson, W. Lawrence, and J. Lopez, Sverdrup Technology, Eglin AFB, FL	AIAA-93-3459 Calculating Crossflow Separation Using Boundary Layer Computations Coupled with an Inviscid Method D. Almosino, Analytical Methods, Redmond, WA	AIAA-93-3460 Roll Damping for Finned Projectiles and Missiles Including Wraparound, Offset, and Arbitrary Number of Fins A. Mikhail, U.S. Army Research Lab., Aberdeen Proving Ground, MD	AIAA-93-3461 Aerodynamic Analysis and Test of a Small-Aspect-Ratio Finned Projectile J. Daywalt and B. Prats, GE Aerospace, King of Prussia, PA						
Session 53-AA-12 —Boundary Layer Aerodynamics						Chaired by: A. Bertlud, NASA Langley, Hampton, VA		San Carlos IV			
AIAA-93-3462 A Laplace Interaction Law for the Computation of Viscous Ailfoil Flow in Low- and High-Speed Aerodynamics F. Arnold, Deutsche Aerospace Airbus, Bremen, Germany; F. Thiele, Technische Universität Berlin, Berlin, Germany	AIAA-93-3463 Boundary Layer Effects on the Flow of a Leading Edge Vortex N. Verhaagen, J. Meeder, and J. Verhelst, Delft Univ., Delft, The Netherlands	AIAA-93-3464 Applications of Generalized Coordinates Boundary-Layer Code to Fortified N.S. Calculations T. Olson and M. Hafez, Univ. of California, Davis, CA; W. Van Dalsem, NASA Ames, Moffett Field, CA	AIAA-93-3465 Development of a System for Transition Characterization A. Bertelrud, AS&M, NASA Langley, Hampton, VA; S. Graves, Lockheed Engineering & Sciences Co., Hampton, VA; J. Diamond and S. Stover, NASA Langley, Hampton, VA	AIAA-93-3466 Development of an Innovative Natural Laminar Flow Wing Concept for High-Speed Civil Transports B. Gibson and H. Gerhardt, Northrop Corp., Pico Rivera, CA	AIAA-93-3467 Application of Natural Laminar Flow to a Supersonic Transport Concept H. Fuhrmann, NASA Langley, Hampton, VA						
Session 12:00 — 2:00 pm AWARDS LUNCHEON Luncheon Speaker: WILLIAM H. DANA, NASA Dryden Flight Research Facility, Edwards, CA										De Anza Ballroom	
"Reflections on an Aerospace Career"											

Mechanics and Control of Flight Award <i>"This award is presented for an outstanding recent technical or scientific contribution by an individual in the mechanics, guidance, or control of flight in space or the atmosphere."</i>			De Florez Training Award for Flight Simulation <i>"This award is named in honor of the late Admiral de Florez and is presented for an outstanding achievement in the application of flight simulation to aerospace training research and development."</i>			Aerodynamics Award <i>"This award is presented for meritorious achievement in the field of applied aerodynamics, recognizing notable contributions in the development, application, and evaluation of aerodynamic concepts and methods."</i>	
---	--	--	--	--	--	--	--

Tuesday Afternoon / August 10, 1993

2:00	2:30	3:00	3:30	4:00	4:30	5:00	5:30
------	------	------	------	------	------	------	------

Session 54-GNC-21—Navigation and Position Location *Chaired by: H. WEISS, Technion-IIT, Haifa, Israel*

Bonsai I

AIAA-93-3819 On Two Misperceptions in the Theory of Inertial Navigation Systems I. Bar-Itzhack, <i>Technion-IIT, Haifa, Israel</i>	AIAA-93-3821 Failure Detection in Hybrid Strapdown-INS/GPS R. Da, <i>Technion-IIT, Haifa, Israel</i>	AIAA-93-3822 On Navigation of a Robotic Land Vehicle B. Friedland and D. Hong, <i>New Jersey Institute of Technology, Newark, NJ</i>	AIAA-93-3823 Error Estimation of Ground Alignment to Arbitrary Azimuth Y. Jiang, <i>Chung Shan Institute of Science & Technology; Y. Lin, National Chiao Tung Univ., Taiwan, ROC</i>	AIAA-93-3825 Autonomous Orbit and Magnetic Field Determination Using Magnetometer and Star Sensor Data M. Psiaki, <i>Cornell Univ., Ithaca, NY</i>			
---	---	---	---	---	--	--	--

Session 55-GNC-22—Space Craft Controls II *Chaired by: F. BAUER, NASA Goddard, Greenbelt, MD*

Bonsai II

AIAA-93-3826 On-Orbit Performance of the Extended Gyrocompass Controller for the TOPEX/POSEIDON Spacecraft P. Sanneman and B. Lee, <i>Fairchild Space & Defense Corp., Germantown, MD</i>	AIAA-93-3827 An Onboard Star Identification Algorithm M. Farniano, <i>NASA Goddard, Greenbelt, MD; K. Ha, Fairchild Space & Defense Corp., Germantown, MD</i>	AIAA-93-3828 Nutation and Precession Control of the High-Energy Solar Physics (HESP) Satellite C. Jayaraman and B. Robertson, <i>McDonnell Douglas Aerospace Systems, Seesbrook, MD</i>	AIAA-93-3829 Modeling of Roll/Pitch Determination with Horizon Sensors for Oblate Earth H. Hablani, <i>Rockwell International, Seal Beach, CA</i>	AIAA-93-3830 The Magellan Aerobraking Experiment: Attitude Control Simulation and Flight Results A. Carpenter, <i>Martin Marietta Astronautics, Denver, CO</i>	AIAA-93-3831 On-Orbit Test of the HST Zero-Gyro Safemode Controller L. Markley, <i>NASA Goddard, Greenbelt, MD; G. Andersen, Jackson & Tull, Seabrook, MD; J. Nelson, Lockheed Missiles & Space Co., Sunnyvale, CA</i>		
--	--	--	--	---	---	--	--

Session 56-GNC-23—GNC Concepts in Air Traffic Control Systems *Chaired by: J. HANSMAN, MIT, Cambridge, MA, and A. MUNDRA, Mitre Corp., McLean, VA*

Bonsai III

AIAA-93-3832 Control Theoretic Approach to Air Traffic Conflict Resolution P. Menón, <i>Optimal Synthesis, Palo Alto, CA</i>	AIAA-93-3833 Development of Advanced Identification Approach and Departure Procedures L. Erkelens, <i>NLR, Amsterdam, The Netherlands</i>	AIAA-93-3834 Statistical Techniques for Traffic Flow Management J. DeArmon, <i>Mitre Corp., McLean, VA</i>	AIAA-93-3835 The Dependent Converging Instrument Approach Procedure A. Smith and A. Mundra, <i>Mitre Corp., McLean, VA</i>	AIAA-93-3836 Overview of the FAA's Differential GPS Cat III/II Feasibility Demonstration Program R. Loh, <i>FAA, Washington, DC; C. Shively, Mitre Corp., McLean, VA</i>			
---	--	---	---	---	--	--	--

Session 57-GNC-24—Flexible Structure Identification *Chaired by: A. CARRIER, Lockheed Research Lab., Palo Alto, CA*

Redwood

AIAA-93-3837 Physical Model-Set Identification for Robust Control of Flexible Structures V. Karlov, R. Glaese, D. Miller, and E. Crawley, <i>MIT, Cambridge, MA</i>	AIAA-93-3838 Linear Systems Identification via Backward-Time Observer J.-N. Juang, <i>NASA Langley, Hampton, VA; M. Phan, Lockheed Engineering & Science Co., Hampton, VA</i>	AIAA-93-3839 Frequency Domain System Identification Methods: Matrix Fraction Description Approach L. Horfa and J.-N. Juang, <i>NASA Langley, Hampton, VA</i>	AIAA-93-3840 Maximum Likelihood Identification and Realization of Stochastic Systems Y. Oshman and T. Mendelboim, <i>Technion-IIT, Haifa, Israel</i>	AIAA-93-3841 Quantification of Model Error as an Interval Model with Nonparametric Error Bounds J.-S. Lew, <i>Tennessee State Univ., Nashville, TN</i>	AIAA-93-3842 Model Identification of Unstable Systems Using ERA Method A. Hu and D. Martin, <i>Dynacs Engineering Co., Palm Harbor, FL</i>		
--	--	---	---	---	---	--	--

Tuesday Afternoon / August 10, 1993 (cont'd)						
2:00	2:30	3:00	3:30	4:00	4:30	5:00
Session 63-FST-10 —Simulation Modeling II <i>Chaired by: J. McCRILLIS, Naval Air Warfare Center, Patuxent River, MD</i>						Los Angeles
AIAA-93-3586 Modeling Design for Aerobraking Orbit Transfer Problem with Simulation <i>Analysis and Study</i> H.-L. Fu, Gardena, CA; R. Culp, Univ. of Colorado, Boulder, CO	AIAA-93-3587 A Radar Altitude and Line-of-Sight Attachment for Flight Simulation R. Sansom and D. Darling, SYRE, Moffett Field, CA	AIAA-93-3588 The Numerical Errors in Inverse Simulation K.-C. Lin, Institute for Simulation & Training, Orlando, FL; P. Lu, Univ. of Iowa, Ames, IA	AIAA-93-3589 A Dual-Euler Method for Solving All-Attitude Angles of the Aircraft X. Huang, China Flight Test Establishment, Xian, PRC	AIAA-93-3590 Initial Development of Research Flight Simulator Software T. Caleffi, Sao Paulo, Brazil		
Session 64-FST-11 —Software Engineering <i>Chaired by: A. KATZ, Univ. of Alabama, Tuscaloosa, AL</i>						Santa Monica
AIAA-93-3591 Enhancing Real-Time Flight Simulation Execution by Replacing Run-Time Library Calls N. Reinbachs, SYRE, Moffett Field, CA	AIAA-93-3592 Multimedia Digital Database Innovations and Application to the Analysis of Weapons Performance G. Butler, Horizons Technology, San Diego, CA; J. Weiler, ONA/SPSD, Alexandria, VA	AIAA-93-3593 An Evaluation of Software Tools for the Design and Development of Cockpit Displays T. Ellis, Unisys Corp., Hampton, VA	AIAA-93-3594 Reusable Code for Helicopter Simulation J. Pickering, Lord Defense Systems, Akron, OH; K.-C. Lin, C. Listle, and G. Guruprasad, Institute for Simulation & Training, Orlando, FL	AIAA-93-3595 Interface of Hardware-in-the-Loop Simulation P.-N. Zhou, Beijing Simulation Center, Beijing, PRC		
Session 65-FST-12 —Simulation Validation and Verification <i>Chaired by: T. GALLOWAY, Naval Training Systems Center, Orlando, FL</i>						San Diego
AIAA-93-3596 Flight Update of Simulator Aerodynamic Math Model K. Neville and T. Stephens, Boeing Commercial Airplane Group, Seattle, WA	AIAA-93-3597 A Flexible Graphical Simulation Monitoring System J. Little, Boeing Commercial Airplane Group, Seattle, WA	AIAA-93-3598 How to Consider Simulation Fidelity and Validity for an Engineering Simulator B. Zhang, China Flight Test Establishment, Xian, PRC	AIAA-93-3599 Simulations and Tests for the Verification of Rendezvous and Docking Systems and Operations W. Fehse and A. Tobias, ESA, Noordwijk, The Netherlands	AIAA-93-3600 Su-27 Flight Dynamics Simulation at High Angle of Attack M. Goman, K. Tatarikov, and A. Khramtsovsky, Central Aerohydrodynamic Institute, Zhukovskiy, Russia		
Session 66-AA-13 —Wing Vortex Aerodynamics <i>Chaired by: L. ERICSSON, Nielsen Engineering & Research, Mountain View, CA</i>						San Carlos I
AIAA-93-3468 A Wind-Tunnel Investigation of the Pressure Distribution on an F1A-13 Wing B. Lee and N. Valerio, National Research Council, Ottawa, Canada	AIAA-93-3469 Flow Visualization of a Rolling Delta Wing and Its Pertinence to the Nonlinear Indicial Response Model G. Addington and J. Jenkins, Wright Lab., Wright-Patterson AFB, OH	AIAA-93-3470 Secondary Flow Control on Slender, Sharp-Edged Configurations T. Ng and D. Gangulee, Univ. of Toledo, Toledo, OH	AIAA-93-3471 Vortex Features of F-106B Aircraft at Subsonic Speeds J. Lamar and J. Brandon, NASA Langley, Hampton, VA; T. Johnson Jr., Lockheed Engineering & Sciences Co., Hampton, VA	AIAA-93-3472 Supersonic Vortex Breakdown over a Delta Wing in Transonic Flow O. Kandil and H. Kandil, Old Dominion Univ., Norfolk, VA; C. Liu, NASA Langley, Hampton, VA	AIAA-93-3473 On the Dynamics of the Juncture Vortex M. Khan, J. Trosper, and A. Ahmed, Texas A&M Univ., College Station, TX	
Session 67-AA-14 —CFD Methods <i>Chaired by: P. RAJ, Lockheed Aeronautical Systems Co., Marietta, GA</i>						San Carlos II
AIAA-93-3474 Computation of Passively Controlled Transonic Wing I. Kim and B. Sung, Korea Aerospace Research Institute, Taejeon, Korea	AIAA-93-3475 Grid and Aerodynamic Sensitivity Analysis of Airplane Components I. Sadreghahighi and R. Smith, NASA Langley, Hampton VA; S. Tiwari, Old Dominion Univ., Norfolk, VA	AIAA-93-3476 Calculation of AGARD Wing 445.6 Flutter Using Navier-Stokes Aerodynamics E. Lee-Rausch and J. Batina, NASA Langley, Hampton, VA	AIAA-93-3478 High-Lift Multiple Element Airfoil Analysis with Unstructured Grids W. Davis, Grumman Aircraft Systems, Bethpage, NY; R. Matus, Fluent, Lebanon, NH	AIAA-93-3479 On the Modeling of Separated Flows about Airfoils P. deMatteis and C. Dima, Capua, Italy		

Session 68-AA-15 —Wind-Tunnel and Design Aerodynamics <i>Chaired by: G. EDWARDS, LORAC Vought Systems Corp., Dallas, TX</i>						San Carlos III	
AIAA-93-3480 Curvature and Leading-Edge Sweepback Effect on Grid Fin Aerodynamic Characteristics M. Miller, Dynetics, Huntsville, AL; W. Washington, U.S. Army Missile Command, Huntsville, AL	AIAA-93-3481 Application of Computational Fluid Dynamics in Aerodynamic Design K. Lee, Univ. of Illinois, Urbana, IL	AIAA-93-3482 A Computational Method for Inverse Design of Transonic Airfoil and Wing X. Zhium, <i>Changsha Institute of Technology, Changsha, PRC</i> ; Z. Ziqiang and W. Liyi, <i>Beijing Univ. of Aeronautics & Astronautics, Beijing, PRC</i>	AIAA-93-3483 Performance Effects of a Large Radome Mounted Aloft a 747-200 Aircraft S. Holcomb, <i>Chrysler Technologies Airborne Systems, Waco, TX</i>	AIAA-93-3484 Design Optimization Study for F-15E Propulsion/Forward Faring Compatibility K. Acheson, S. Lehman, and T. Smith, <i>McDonnell Douglas Aerospace, St. Louis, MO</i>	AIAA-93-3485 Aerodynamic Investigation with Focusing Schlieren in a Cryogenic Wind Tunnel E. Gartenberg, L. Weinstein, and E. Lee Jr., <i>Old Dominion Univ., Norfolk, VA</i>		
Session 69-AA-16—Supersonic/Hypersonic Aerodynamics <i>Chaired by: W. BROOKS, LORAC Vought Systems Corp., Dallas, TX</i>						San Carlos IV	
AIAA-93-3486 A Computational Study of Hypersonic Boundary Layer Stability on a Blunt Nose Flare Cone Model N. Chokani, North Carolina State Univ., Raleigh, NC; S. Wilkinson, NASA Langley, Hampton, VA	AIAA-93-3487 Non slender Waveriders X. He and G. Emanuel, <i>Univ. of Oklahoma, Norman, OK</i>	AIAA-93-3488 Computational Analysis of Off-Design Waveriders X. He and M. Rasmussen, <i>Univ. of Oklahoma, Norman, OK</i>	AIAA-93-3489 Flowfield Calculations over a Swept Wing with a Gap in High Enthalpy Air D. Mrozinski, <i>Wright Lab., Wright-Patterson AFB, OH</i>	AIAA-93-3490 A Three-Dimensional Supersonic Navier-Stokes Solution over an F/A-18 C/D Forebody with Inlet Bleed W. Dwyer and J. Math, <i>Northrop B-2 Division, Pico Rivera, CA</i>	AIAA-93-3491 Preliminary Design Estimates of High-Speed Streamlines on Arbitrary Shaped Vehicles Defined by Quadrilateral Elements K. Dieters, <i>McDonnell Douglas Corp., St. Louis, MO</i>		

Wednesday Morning / August 11, 1993							
9:00	9:30	10:00	10:30	11:00	11:30	12:00	12:30

Session 70-GNC-27—Spacecraft Guidance and Navigation II <i>Chaired by: J. SUNKEL, NASA Johnson, Houston, TX</i>						Bonsai I	
AIAA-93-3854 Autonomous Star Field Identification for Solar System Exploration M. Scholl, <i>Alenka Associates, Tempe, AZ</i>	AIAA-93-3855 Computation of Optimal Low- and Medium-Thrust Orbit Transfers C.-H. Chuang and T. Goodson, <i>Georgia Institute of Technology, Atlanta, GA</i> ; J. Hanson, <i>NASA Marshall, Huntsville, AL</i>	AIAA-93-3856 Rendezvous Navigation for a Two-Spacecraft Stable Orbit Rendezvous in a Small Radium Translunar Halo Orbit B. Jones and R. Bishop, <i>Univ. of Texas, Austin, TX</i>	AIAA-93-3857 Decentralized Autonomous Attitude Determination Using an Internally Stabilized Payload Y. Oshman and M. Isakow, <i>Technion-IT, Haifa, Israel</i>	AIAA-93-3858 A Hybrid Approach to Near-Optimal Atmospheric Guidance for Aerassisted Orbit Transfer Maneuvers M. McFarland and A. Calise, <i>Georgia Institute of Technology, Atlanta, GA</i>	AIAA-93-3860 Analysis of the Targeting Error Induced by Hill's Guidance H. Iida, <i>NEC Corp., Yokohama, Japan</i> ; R. Bishop, <i>Univ. of Texas, Austin, TX</i>		
Session 71-GNC-28—Space Automation and Robotic Control <i>Chaired by: C. JACOBUS, Cybernet Systems Corp., Ann Arbor, MI</i>						Bonsai II	
AIAA-93-3861 Intelligent Teach-In Procedure for Multirobot Programming through Three-Dimensional Measurement Y. Matsuka and H. Tsai, <i>Univ. of Kansas, Lawrence, KS</i>	AIAA-93-3862 Attitude Control of Space Robot with Flexible Appendage H. Fujii and I. Murayama, <i>Tokyo Metropolitan Institute of Technology, Tokyo, Japan</i>	AIAA-93-3863 Theoretical and Experimental Studies for Continuous Path Control of Flexible Manipulators Mounted on a Free-Flying Space Robot Y. Murotsu, K. Senda, A. Mitsuya, K. Yamane, and M. Hayashi, <i>Univ. of Osaka Prefecture, Osaka, Japan</i> ; T. Nunohara, <i>Kyokuto Kahtatsu Kogyo Co., Hyogo, Japan</i>	AIAA-93-3864 Motion-Planning of a Dual Arm Free-Floating Manipulator with Inertially Fixed Base S. Agrawal and S. Shurumalla, <i>Ohio Univ., Athens, OH</i>	AIAA-93-3865 Sliding Mode End Point Trajectory Control and Stabilization of Elastic Multibody System S. Chennamalla and S. Singh, <i>Univ. of Nevada, Las Vegas, NV</i>	AIAA-93-3866 A Space Manipulator with Inertially Fixed Base? M. Stieber and P. Nguyen, <i>Canadian Space Agency, Ontario, Canada</i>	AIAA-93-3867 A Passivity Based Controller of Free-Base Manipulators C.-H. Chuang and M. Mital, <i>Georgia Institute of Technology, Atlanta, GA</i> ; J.-N. Juang, <i>NASA Langley, Hampton, VA</i>	

Session 72-GNC-29—Control Applications of AI <i>Chaired by: J. HARRISON, Charles Stark Draper Lab., Cambridge, MA</i>							Bonsai III
AIAA-93-3868 Fuzzy Logic Control Algorithm for Suppressing E-6A Long Trailing Wire Antenna Wind Shear Induced Oscillations R. Bort, G. Griesz, and A. Ouyne, Boeing Defense & Space Group, Seattle, WA	AIAA-93-3869 Aircraft Failure Detection and Identification Using Neural Networks M. Napolitano, S. Naylor, and C. Chen, West Virginia Univ., Morgantown, WV	AIAA-93-3870 Fault Detection, Isolation, and Reconfiguration for Aircraft Using Neural Networks R. Kline-Schoder and H. Rauch, Lockheed Research Lab., Palo Alto, CA; H. Youssef, Lockheed Aeronautical Systems Co., Marietta, GA	AIAA-93-3871 Vision-Based Obstacle Detection and Grouping for Helicopter Guidance B. Snidhar and G. Chatterji, NASA Ames, Moffett Field, CA	AIAA-93-3872 Neural On-Line Learning in Missile Guidance J. Dalton and S. Balakrishnan, Univ. of Missouri, Rolla, MO	AIAA-93-3873 Multiple Near-Optimal Solutions for a Structural Control Problem Using a Generic Algorithm with Niching K. Krishnakumar, R. Swaminathan, and L. Montgomery, Univ. of Alabama, Tuscaloosa, AL		
Session 73-GNC-30—Flexible Structure Control Applications <i>Chaired by: L. HORTA, NASA Langley, Hampton, VA</i>							Redwood
AIAA-93-3874 Robust Identification and Vibration Suppression of a Flexible Structure J. Crassidis, D. Leo, and J. Mook, SUNY, Buffalo, NY; D. Inman, Virginia Tech, Blacksburg, VA	AIAA-93-3875 Human-in-the-Loop Evaluation of RMS Active Damping Augmentation M. Demeo, Vigen, Hampton, VA; M. Gilbert and M. Scott, NASA Langley, Hampton, VA; J. Lapenle, Charles Stark Draper Lab., Cambridge, MA; E. Bains, NASA Johnson, Houston, TX; M. Jensen, Lockheed Engineering & Sciences Co., Houston, TX	AIAA-93-3876 Experiments in Modeling and Control of the ASCIE Segmented Reflector A. Carrier and J.-N. Auburn, Lockheed Research Lab., Palo Alto, CA	AIAA-93-3877 Adaptive Neurocontrol for Large Flexible Structures K. Krishnakumar and L. Montgomery, Univ. of Alabama, Tuscaloosa, AL	AIAA-93-3878 A Layered Approach to Structural Control System Design for JPL's CSI Phase-1 Model C.-C. Chu and J. O'Brian, JPL, Pasadena, CA	AIAA-93-3879 Robust Controllers for the Middeck Active Control Experiment Using Popov Control J. How and S. Hall, MIT, Cambridge, MA		
Session 74-GNC-31—Nonlinear Control Applications <i>Chaired by: C. HUANG, Grumman Corp., Bethpage, NY</i>							Ironwood
AIAA-93-3880 Approximate Decoupling Flight Control System Design with Output Feedback for Nonlinear System S. Park and M. Nagati, Wichita State Univ., Wichita, KS	AIAA-93-3881 A New Technique for Nonlinear Control of Aircraft A. Khan and P. Lu, Iowa State Univ., Ames, IA	AIAA-93-3882 Autopilot Design for a Bank-to-Turn Missile Using the Nyquist Array Method J. Zheng, Z. Luan, P. Cui, and D. Yang, Harbin Institute of Technology, PRC; W. Feng, P. Xu, and H. Yang, Peking Institute of Electronic System Engineering, PRC	AIAA-93-3883 Genetic Design of Digital Model-Following Flight Control System B. Porter and D. Morris, Univ. of Salford, Salford, UK	AIAA-93-3884 Prediction and Control of Limit Cycling Behavior in Boosting Rockets B. Newman, Chandler, AZ	AIAA-93-3886 Multivariable Control of the Space Shuttle Remote Manipulator System Using Linearization by State Feedback C. Gertman and N. Adams, and M. Bedrossian, Charles Stark Draper Lab., Cambridge, MA; Seabrook, TX; L. Valavan, MIT, Cambridge, MA		
Session 75-GNC-32—Missile Midcourse Guidance and Navigation <i>Chaired by: W. ARNOLD III, Sverdrup Technology, Eglin AFB, FL</i>							Cottonwood
AIAA-93-3887 The Use of GPS for Evaluating Inertial Measurement Unit Errors J. Dougherty, H. El-Sheniel, and D. Hohman, TRW Ballistic Missiles Division, San Bernardino, CA	AIAA-93-3888 Guidance Law Based on Piecewise Constant Control for Hypersonic Gliders D. Hull, Univ. of Texas, Austin, TX; J.-M. Seguin, Aerospatiale, Paris, France	AIAA-93-3889 Monte Carlo Analysis of the Titan II/ITOS Guidance System for the Mars Observer Mission S. Bell, M. Ginsburg, and P. Rao, Martin Marietta Corp., Denver, CO	AIAA-93-3890 Spacecraft Intercept Guidance Using Zero Effort Miss Steering B. Newman, Orbital Sciences Corp., Chandler, AZ	AIAA-93-3891 A Study of Recent Strapdown Navigation Attitude Algorithms J. Murphy and H. Musoff, Charles Stark Draper Lab., Cambridge, MA	AIAA-93-3892 Transfer Alignment Design and Evaluation C. Yang and C.-F. Lin, American GNC Corporation, Chatsworth, CA; D. Tarrant, C. Roberts, P. Rutlin, and T. Hester, U.S. Army Missile Command, Redstone Arsenal, AL	AIAA-93-3893 Analytical Derivation of Trajectory Shaping Guidance Q. Wang and C. Lin, American GNC Corp., Chatsworth, CA; C. D'Souza, Wright Lab., Eglin AFB, FL	

Session 76-AFM-16—Unsteady Aerodynamics II <i>Chaired by:</i> N. CHADERJIAN, NASA Ames, Moffett Field, CA						Colton I&I	
AIAA-93-3682 Nonlinear and Unsteady Aerodynamic Responses of a Rolling 65-Degree Delta Wing (Invited) J. Jenkins, Wright Lab., Eglin AFB, FL, E. Hanil, National Research Council, Ontario, Canada	AIAA-93-3683 Flow Physics of Critical State for Rolling Delta Wings I. Ericsson, Mountain View, CA	AIAA-93-3684 Wing Rock of Delta Wings with the Analysis by the Phase Plane Method T. Yoshiyagi, A. Tate, and J. Noda, Supersonic Windtunnel Control Lab., Tokyo, Japan	AIAA-93-3685 The Control of Wing Rock by Forebody Blowing, Z. Celik, L. Roberts, and N. Pedreiro, Stanford Univ., Stanford, CA	AIAA-93-3686 Prediction of Static and Dynamic Airloads on a Delta Wing at High Incidence X. Huang and E. Hanifi, IAR/NRC, Ontario, Canada	AIAA-93-3687 Navier-Stokes Computations on a Full-Span Wing-Body Configuration with Oscillating Control Surfaces S. Obayashi, T. Chiu, and G. Guruswamy, NASA Ames, Moffett Field, CA	AIAA-93-3688 Simulation of Tall Buffet Using Delta Wing-Vertical Tail Configuration O. Kandil, H. Kandil, and S. Massey, Old Dominion Univ., Norfolk, VA	
Session 77-AFM-17—Re-Entry Technology <i>Chaired by:</i> G. CHRUSCIEL, Lockheed Missiles & Space Co., Sunnyvale, CA						Ferrante I	
AIAA-93-3689 Uranus and Neptune Atmospheric Entry Probe Study M. Tauber and P. Wercinski, NASA Ames, Moffett Field, CA; L. Yang, Sterling Software, Palo Alto, CA; J. Paterson, Mountain View, CA	AIAA-93-3690 Investigation of the Launch Pad Abort Capabilities of the HL-20 Lifting Body B. Jackson and R. Rivers, NASA Langley, Hampton, VA; R. Chowdry, Lockheed Engineering & Sciences Co., Hampton, VA; W. Ragsdale and D. Geyer, UNISYS Corp., Hampton, VA	AIAA-93-3691 Optimal Return to Launch-Site Abort Trajectories for an HL-20 Personnel Launch Vehicle K. Dutton, NASA Langley, Hampton, VA	AIAA-93-3692 CFD Predictions of Flowfields about a Hypersonic Lifting Body A. Le, T. Shivananda, and T. Holtz, TRW Ballistic Missiles Division, San Bernardino, CA; R. Marx, USAF, Kirtland AFB, NM	AIAA-93-3693 COMET Recovery System Flight Dynamics S. Hill and T. McCusker, Space Industries, League City, TX	AIAA-93-3694 Ascent Abort Capability for a Two-Stage Fully Reusable Advanced Manned Launch System J. Nathel, D. Stanley, and W. Engelund, NASA Langley, Hampton, VA	AIAA-93-3695 Landing Dispersion Analysis for the COMET Recovery System T. McCusker and S. Hill, Space Industries, League City, TX	
Session 78-FST-13—Pilot Training <i>Chaired by:</i> E. AIKEN, NASA Ames, Moffett Field, CA						Santa Monica	
AIAA-93-3601 Future Military Pilot Training—A Perspective of Industry D. Illauer, Deutsche Aerospace, Munich, Germany	AIAA-93-3602 The Future of Flying Work Modeling for Pilot Educational Tasks V. Ponomarenko, Institute of Aviation & Space Medicine, Moscow, Russia	AIAA-93-3603 Computerized Learning of Pilots to Spatial Orientation Flight Tasks V. Ponomarenko, A. Vorona, D. Gander, and V. Ussov, Institute of Aviation & Space Medicine, Moscow, Russia					
Session 79-FST-14 —In-Flight Simulation <i>Chaired by:</i> V. LEBACQZ, NASA Ames, Moffett Field, CA						San Diego	
AIAA-93-3604 USAF In-Flight Simulation: A Cost-Effective Operating Approach S. Markman, Wright Lab., Wright-Patterson AFB, OH	AIAA-93-3605 Flight and Ground Simulation Evaluation of the Proposed USAF Head-Up Display Standard R. Bailey and L. Knotts, Calspan Corp., Buffalo, NY	AIAA-93-3606 A Rapid Prototyping System for In-Flight Simulation Using the Calspan Learjet 25 S. Buehler and P. Depege, Calspan Corp., Buffalo, NY					
Session 80-FST-15—Design & Certification Standards in Flight Simulation <i>Chaired by:</i> C. LAPISKA, CAE-Link Corp., DFW Airport, TX						Los Angeles	
			Panel Discussion Panel Participants E. Cook, FAA, Atlanta, GA; R. Foster, USAir, Pittsburgh, PA; J. DePaola, Consultant, Euless, TX; R. Hess, Simtec Inc, Manassas, VA				

Wednesday Morning / August 11, 1993 (cont'd)

9:00	9:30	10:00	10:30	11:00	11:30	12:00	12:30
Session 81-AA-17 —High Alpha Aerodynamics <i>Chaired by:</i> W. ELY, McDonnell Aircraft Co., St. Louis, MO							San Carlos I
AIAA-93-3492 Effect of Leeward Flow Dividers on the Wing Rock of a Delta Wing T. Ng, J. Kountz, T. S. Skaff, and B. Lowery, <i>Univ. of Toledo, Toledo, OH</i>	AIAA-93-3493 Aerodynamic Characteristics of the MMPT ATD Vehicle at High Angles of Attack E. Smith and M. Salezar, <i>Naval Warfare Center, China Lake, CA; S. Hebbard and M. Platzer, Naval Postgraduate School, Monterey, CA</i>	AIAA-93-3494 Flow Control over Delta Wings at High Angles of Attack S. Kluge, O. Radinolis, and D. Telonis, <i>Virginia Tech, Blacksburg, VA</i>	AIAA-93-3495 Numerical Simulation of the Unsteady Flows about an F-18 Aircraft in the High-Alpha Regime S. Murman, L. Schiff, and Y. Rizk, <i>MCAT Institute, Moffett Field, CA</i>	AIAA-93-3496 An Experimental Study of Droop Leading Edge Modifications on High- and Low-Aspect-Ratio Wings up to 50-Degrees Angle of Attack H. Gonzalez, <i>Naval Air Warfare Center, Warminster, PA; A. Winkelmann, Univ. of Maryland, College Park, MD</i>	AIAA-93-3497 High-Angles-of-Attack Aerodynamics of Delta-Wings Planform Using "Pop Up" Vortex Generators D. Manor, <i>Parks College of St. Louis Univ., Cahokia, IL</i>		
Session 82-AA-18 —Configuration Aerodynamics <i>Chaired by:</i> W. MASON, Virginia Tech, Blacksburg, VA							San Carlos II
AIAA-93-3537 Application of a Parabolized Navier-Stokes Code to an HSCT Configuration and Comparison to Wind-Tunnel Data D. Hollenback and G. Blom, <i>Boeing Commercial Airplane Group, Seattle, WA</i>	AIAA-93-3499 A Flowfield Study of a Close-Coupled Canard Configuration R. Howard and J. O'Leary, <i>Naval Postgraduate School, Monterey, CA</i>	AIAA-93-3500 Wing Pylon Fillet Design Using Unstructured Mesh Euler Solvers M. Potsdam and G. Inlemann, <i>McDonnell Douglas Aerospace, Long Beach, CA; N. Frink, R. Campbell, and L. Smith, NASA Langley, Hampton, VA; S. Pirzadeh, Vigney Inc., Hampton, VA</i>	AIAA-93-3501 Wing Planforms Optimized to Reduce the Center-of-Pressure Shift from Subsonic to Supersonic Mach Numbers S. Elkayam and A. Sigal, <i>Haifa, Israel</i>				
Session 83-AA-19 —Wing/Airfoil Aerodynamics <i>Chaired by:</i> D. MUILENBURG, Boeing Co., Seattle, WA							San Carlos III
AIAA-93-3502 Prediction of the Stall and Post-Stall Behavior of Airfoils at Low and High Reynolds Numbers T. Cebeci, <i>California State Univ., Long Beach, CA; L. Carr, NASA Ames, Moffett Field, CA</i>	AIAA-93-3503 Numerical Simulation of Incompressible Viscous Flow around a Marine Propeller W-G. Park and L. Sankar, <i>Georgia Institute of Technology, Atlanta, GA</i>	AIAA-93-3504 Lift-Enhancing Tab for Multi-Element Airfoil J. Ross, B. Storms, and P. Carrannano, <i>NASA Ames, Moffett Field, CA</i>	AIAA-93-3505 Computational Study of a Conical Wing Having Unit Aspect Ratio at Supersonic Speeds B. McGrath, <i>Lockheed Engineering & Sciences Co., Hampton, VA</i>	AIAA-93-3506 Using Surface Transpiration with an Euler Method for Cost-Effective Aerodynamic Analysis P. Raj and B. Harris, <i>Lockheed Aeronautical Systems Co., Marietta, GA</i>	AIAA-93-3507 Aerodynamic Characteristics of an External Store Carriage: Part A O. Ozcan, <i>Istanbul Technical Univ., Istanbul, Turkey</i>		
Session 84-AA-20 —Vortical/Vortex Aerodynamics <i>Chaired by:</i> S. SCHRECK, USAF Academy, Colorado Springs, CO							San Carlos IV
AIAA-93-3508 Experimental and Computational Investigations of the Flowfield Around the F117A S. Vermeersch, D. Modiano, I. Harisopoulos, and E. Murman, <i>MIT, Cambridge, MA; J. Peraire, Imperial College, London, UK</i>	AIAA-93-3509 Computation of Wake Roll-up for Complete Aircraft Configurations R. Ribeiro and O. Resende, <i>Embraer S.A., São Paulo, Brazil</i>	AIAA-93-3510 The Application of an Euler and a Navier-Stokes Method to the Vortical Flow about a Delta Wing F. Brandsma, S. Bosse, H. Hoeijmakers, <i>National Aerospace Lab., Amsterdam, The Netherlands; J. van den Berg, Technical University Delft, The Netherlands</i>	AIAA-93-3511 Attenuation of Airplane Wake Vortices by Excitation of Far-Field Instability V. Nikolic and E. Jumper, <i>Univ. of Notre Dame, Notre Dame, IN</i>	AIAA-93-3512 Buret-Vortex Flowfield Unsteadiness F. Roos, <i>McDonnell Douglas Corp., St. Louis, MO</i>	AIAA-93-3513 Lift Augmentation Via Vortex Trapping on Some Delta Wings M. Buckholz and J. Tso, <i>California Polytechnic State Univ., San Luis Obispo, CA</i>		

Wednesday Afternoon / August 11, 1993

2:00	2:30	3:00	3:30	4:00	4:30	5:00	5:30
Session 85-AA-21 —Experimental Aerodynamics <i>Chaired by: M. McVEIGH, Boeing Helicopters, Philadelphia, PA</i>						San Carlos I	
AIAA-93-3514 Experimental Analysis of Rotary Derivatives on a Modern Aircraft Configuration G. Guglieri and F. Quagliotti Politecnico di Torino, Torino, Italy	AIAA-93-3515 Laser Holographic Interferometric Measurements of the Flow behind a Rearward Facing Step R. Leonard and N. Chokani, North Carolina State Univ., Raleigh, NC	AIAA-93-3516 Small-Scale Wind-Tunnel Investigation of an Advanced Fighter Configuration Semiplan Wing S. Harris, NASA Ames, Moffett Field, CA; K. Langan and K. Iwanski, Wright Lab., Wright-Patterson AFB, OH	AIAA-93-3517 Flow Visualization of Mast-Mounted-Sight/Main Rotor Aerodynamic Interactions T. Ghese, Analytical Services & Materials, Hampton, VA; H. Kelley, NASA Langley, Hampton, VA	AIAA-93-3518 Measurements in 80-by-120-Foot Wind Tunnel of Hazard Posed by Lift-Generated Wakes V. Rossow, J. Sacco, and P. Askins, NASA Ames, Moffett Field, CA	AIAA-93-3519 Full-Scale Wind Tunnel Studies of F/A-18 Tall Buffet L. Meyn, NASA Ames, Moffett Field, CA; K. James, Sterling Federal Systems, Moffett Field, CA		
Session 86-AA-22 —CFD—Aerodynamics <i>Chaired by: F. WITZEMAN JR., Wright Lab., Wright-Patterson AFB, OH</i>						San Carlos II	
AIAA-93-3520 Semidiscrete Galerkin Solution of the Compressible Boundary Layer Equations with Viscous-Inviscid Interaction B. Day and A. Meade Jr., Rice Univ., Houston, TX	AIAA-93-3521 A Three-Dimensional Navier-Stokes Analysis of a Generic Ground-Vehicle Shape M. Malone, W. Dwyer, and Richard Crouse, Northrop B-2 Division, Pico River, CA	AIAA-93-3522 Integrated Jet Interaction CFD Predictions and Comparison to Force and Movement Measurements for a Thruster Attitude Controlled Supersonic Missile S. Chan and R. Roger, Teledyne Brown Engineering, Huntsville, AL	AIAA-93-3523 A Pointwise Version of the Baldwin-Barth Turbulence Model U. Goldberg, Rockwell International Science Center, Thousand Oaks, CA	AIAA-93-3524 Effects of Turbulence Models and Adaptive Grids on the Jet Interaction Flowfield S. Clark, Mevatec Corp., Huntsville, AL; S. Chan, Teledyne Brown Engineering, Huntsville, AL	AIAA-93-3525 UNISG: An Interactive Multizone Structure Grid Generator K-Y. Szema, S. Ramakrishnan, D. Ota, and C-L. Chen, Rockwell International Science Center, Thousand Oaks, CA		
Session 87-AA-23 —VSTOL/STOL Aerodynamics <i>Chaired by: M. SUMMA, Analytical Methods, Inc., Redmond, WA</i>						San Carlos III	
AIAA-93-3526 Characteristics of Deformable Leading-Edge for High-Performance Helicopter Rotor S. Lee, K. McAlister, and C. Tung, NASA Ames, Moffett Field, CA	AIAA-93-3527 Navier-Stokes Calculations of Rotating BERP Platform Blade Flowfields C. Nam and D-H. Lee, Seoul National Univ., Seoul, Korea	AIAA-93-3528 Numerical Vorticity Capturing for Helicopter Rotor and Body Flows C. Wang, Georgia Institute of Technology, Atlanta, GA; J. Steinhoff, Univ. of Tennessee Space Institute, Tullahoma, TN; J. Bridgman, Flow Analysis, Tullahoma, TN	AIAA-93-3530 Calculation of V/STOL Aircraft Aerodynamics with Deflected Jets in Ground Effect P. Liaw and Y-S. Chen, Huntsville, AL; E. Lan, Univ. of Kansas, Lawrence, KS	AIAA-93-3531 A Closed-Form, Approximate Analytical Solution for Rotor/Wing Interactions in Hover L. Young, NASA Ames, Moffett Field, CA; K. Derby, Sterling Software, Moffett Field, CA	AIAA-93-3532 Numerical Flowfield Simulation of the V-22 Tiltrotor Airframe in Forward Flight V. Raghaven, Sterling Software, Moffett Field, CA; W. McCroskey, NASA Ames, Moffett Field, CA		
Session 88-AA-24 —High-Lift Aerodynamics <i>Chaired by: N. WOOD, Univ. of Bath, Bath, UK</i>						San Carlos IV	
AIAA-93-3533 Two-Dimensional Computational Analysis of a Transport High-Lift System and Comparison with Flight-Test Results J. Hardin, Lockheed Engineering & Sciences Co., Hampton, VA; C. van Dam, Univ. of California, Davis, CA; L. Yip, NACA Langley, Hampton, VA	AIAA-93-3534 Navier-Stokes Applications to High-Lift Aircraft Analysis W. Valarezo, McDonnell Douglas Aerospace, Long Beach, CA; D. Mavriplis, ICASE, Hampton, VA	AIAA-93-3535 PMARC Analysis of a Subsonic Transport High-Lift Configuration H. Pauley and J. Perkins, North Carolina State Univ., Raleigh, NC; J. Hardin and L. Yip, NASA Langley, Hampton, VA	AIAA-93-3536 Three-Dimensional Subsonic Aerodynamic Analysis of a High-Lift Transport Configuration S. Dobbela, Vigyan, Hampton, VA	AIAA-93-3538 The Experimental Investigation of Lift Enhancement Produced by Oscillating a Trailing Edge Flap of a Solid Wing S. Greenhalgh, Naval Air Warfare Center, Warminster, PA			

TABLE OF CONTENTS

Flight Simulation History (12-FST-1)

93-3545	Evolution of Flight Simulation L.Allen.....	1
93-3546	Technology Evolution of Flight Simulation G.Abbey	N/A
93-3547	Lessons Learned from a Historical Review of Piloted Aircraft Simulators at NASA Ames Research Center S.Anderson and R.Anderson.....	N/A
93-3548	Internally Coherent System of Innovation: The Case of Flight Simulation R.Miller and F.Olleror.....	12

Simulation Modeling I (13-FST-2)

93-3549	Development and Validation of a Blade Element Mathematical Model for the AH-64 Apache Helicopter H.Mansur	N/A
93-3550	An Advanced Rotorcraft Flight Simulation Model: Parallel Implementation and Performance Analysis S.Sarathy and V.Murthy	29
93-3551	Methodology for Integration of Digital Control Loaders in Aircraft Simulators A.Katz and M.Scharle.....	36
93-3552	Dynamic Simulation Fidelity Improvement Using Transfer Function State Extrapolation W.Bezdek and L.Moody.....	44
93-3553	Aircraft Threat Modeling from Performance Data M.Anderson and D.Schab.....	53
93-3554	Universal Controls Loading System L.Zaychik, A.Predtechensky, V.Rodchenko, and Y.Yashin.....	N/A

Simulation Facilities I (14-FST-3)

93-3555	Development of a Graphical User Interface to Facilitate Development and Certification of a New Commercial Airplane J.Little.....	N/A
93-3556	D3S: The Delphins Display Design System E.Theunissen	62
93-3557	The Flight Simulation Technology for Integrated Flight/Trajectory Control System K.Wang.....	N/A
93-3558	A High Fidelity Video Delivery System for Real-time Flight Simulation Research D.Wilkins and C.Roach.....	68

93-3559	Hybrid Complex of the Aircraft Intellectualized Control Systems Simulation at the Stage of Their Research Projecting A.Yakovlev and V.Vasilets	74
---------	---	----

Human Factors and Pilot Cueing (30-FST-4)

93-3560	Texture as a Visual Cueing Element in Computer Image Generation.Part I Representation of the Sea Surface J.Sinacori	341
93-3561	What Optical Cues Do Pilots Use to Initiate the Landing Flare? Results of a Piloted Simulator Experiment S.Advani, J.van der Vaart, R.Rysdyk and J.Grosz	81
93-3562	Acoustical and Vibratory Stimuli Interdependencies and their Applications in Simulation and Cue Synchronization R.Sawler and R.Matusof	90
93-3563	Transport Delay Compensation: An Inexpensive Alternative to Increasing Image Generator Update Rate F.Cardullo and G.George	95
93-3564	Multistage Integration Model for Human Egomotion Perception G.Zacharias , A.Miao and R. Warren	103
93-3565	Technique Problems of Visual and Motion Cuing Simulation V.Rodchenko and A.Predtechensky	N/A

Modeling the Simulation Environment (31-FST-5)

93-3566	Visual Weather Simulation Using Meteorological Databases B.Montag	114
93-3567	Line of Sight Determination in Real-Time Simulations F.Kull Jr. and D.Fought	123
93-3568	The Challenges of Simulating Wake Vortex Encounters and Assessing Separation Criteria E.Dunham, R.Stuever, and D.Victory	N/A
93-3607	Terrain Modeling for Real-Time Photo Texture Based Visual Simulation V.Devarajan and D.McArthur	133

Simulation Applications (32-FST-6)

93-3569	Fly-by-Throttle Transport Simulation Controller Comparison Study J.Conley	N/A
93-3570	A Primary Flight Display for Four-Dimensional Guidance and Navigation Influence of Tunnel Size and Level of Additional Information on Pilot Performance and Control Behaviour E.Theunissen	140
93-3571	The Use of a Simulator in Aircraft Accident Investigation T.Nieuwpoort and J.van Hengst	N/A

93-3572	Prognostic Evaluation of Getting Trauma Induced by Impact Accelerations U.Mazurin and G.Stypakov	324
93-3608	A Comparative Evaluation of Three Take-Off Performance Monitor Display Types J.Verspay and R. Khatwa.....	147

Simulation Facilities 2 (47-FST-7)

93-3573	The Advanced Concepts Flight Simulator: A Full-Mission Simulation Research Tool for Human Factors B.Sullivan, R.Shiner, and R.Panda.....	N/A
93-3574	The Development of a Simulator Facility for Advanced Research Into Simulation Techniques, Motion System Control and Navigation Systems Technologies S Advani	156
93-3575	Development and Operationsof a Real-Time Simulation at the NASA Ames Vertical Motion Simulator C.Sweeney, S.Sheppard and M. Chetelat.....	167
93-3576	An Asynchronous Simulation Methodology for Improving the Performance of Training Simulators W.Deiss, E.Fadden and R.Howe.....	174
93-3577	Semi-Full-Scale Dynamic Simulation Complex on the Basis of Centrifuge V.Vasiletz and O.Yakimenko.....	187

Motion Systems (48-FST-8)

93-3578	False Cue Detection Thresholds in Flight Simulation R.Hosman and H.van der Steen.....	193
93-3579	Simulation Motion Effects on Single Axis Compensatory Tracking J.Schroeder	202
93-3580	Pilot Evaluations of Augmented Flight Simulator Motion S.Garrood and L.Reid	214
93-3581	New Concept of the Motion System for the Low-Cost Flight Simulator Development and Design V.Shibaev	N/A

Simulation Networking (49-FST-9)

93-3582	Representation of Vehicle Location in Networked Simulations K-C.Lin, B.Goldiez and H.Ng	222
93-3583	Implementation of Expert Systems within an Interactive Tactical Environment D.Siksik and R.Rourke	227
93-3584	Dissimilar Computer Image Generators Provide Challenge in Development of Apache Longbow Full-Mission Engineering Simulator J.Richard.....	N/A
93-3585	Pseudo Aircraft Systems: A Multi-Aircraft System for Air Traffic Control Research R.Weske and G.Danek	234

Simulation Modeling II (63-FST-10)

93-3586	Modeling Design for Aerobraking Orbit Transfer Problem with Simulation Analysis and Study H-L.Fu and R.Culp	N/A
93-3587	A Radar Altitude and Line-of-Sight Attachment R.Sansom and D.Darling	243
93-3588	The Numerical Errors in Inverse Simulation K-C.Lin, P.Lu and M. Smith	251
93-3589	A Dual-Euler Method for Solving All-Attitude Angles of the Aircraft H.Xueqiao	257
93-3590	Initial Development of a Research Flight Simulator Software T.Caleffi and E.Belo	263

Software Engineering (64-FST-11)

93-3591	Enhancing Real-Time Flight Simulation Execution by Intercepting Run-Time Library Calls N.Reinbachs	267
93-3592	Multimedia Digital Database Innovations and Application to the Analysis of Weapons Performance J.Welter and G.Butler	N/A
93-3593	An Evaluation of Software Tools for the Design and Development of Cockpit Displays T.Ellis	275
93-3594	Reusable Code for Helicopter Simulation J.Pickerine, K-C.Lin, C.Lisle, and G.Guruprasad	281
93-3595	Interface of Hardware-in-the-Loop Simulation P-N.Zhou	N/A

Simulation Validation and Verification (65-FST-12)

93-3596	Flight Update of Simulator Aerodynamic Math Model K.Neville and T.Stephens	289
93-3597	A Flexible Graphical Simulation Monitoring System J.Little	N/A
93-3598	How to Consider Simulation Fidelity and Validity for an Engineering Simulator B.Zhang	298
93-3599	Simulations and Tests for the Verification of Rendezvous and Docking Systems and Operations W.Fehse and A.Tobias	N/A
93-3600	Su-27 Flight Dynamics Simulation at High Angle of Attack M.Goman, K.Tatamikov, and A.Khrantsovsky	N/A

Pilot Training (78-FST-13)

93-3601	Future Military Pilot Training: A Perspective of Industry D.Illauer.....	306
93-3602	The Future of Flying Work Modeling for Pilot Educational Tasks V.Ponomarenko.....	N/A
93-3603	Computerized Teaching of Pilots to Spatial Orientation Flight Tasks V.Ponomarenko, A.Vorona, D.Gander, and V.Ussov	337

In-Flight Simulation (79-FST-14)

93-3604	USAF In-Flight Simulation: A Cost-Effective Operating Approach S.Markman.....	316
93-3605	Flight and Ground Simulation Evaluation of the Proposed USAF Head-Up Display Standard R.Bailey and L.Knotts	N/A
93-3606	A Rapid Prototyping System for In-Flight Simulation Using the Calspan Learjet 25 S.Buethel and P.Deppe	320

EVOLUTION OF FLIGHT SIMULATION

L. D. Allen*
CAE Electronics
Montreal, Canada

Abstract

The idea of learning to fly while still safely on the ground goes back as far as the Wright Brothers. However, it was not until Ed Link designed his famous Blue Box in 1929, that this dream became a possibility. Although initial acceptance was slow, it finally sold in large numbers and the flight simulation industry was launched. It was 14 years later in 1943, that the first electronic flight trainer appeared, built by Bell Telephone Labs for the US Navy's PBM-3 aircraft.

This paper traces the development of flight simulation from these early days to the present, including the development of standards for simulator approval and the IATA data requirements.

Introduction

Flight simulation technology has undergone a remarkable evolution during the last 85 years. While it is not possible to give a full account of all of the many inventions that have been tried, I have included the major highlights, with more emphasis on developments of the last 4 decades, biased somewhat by my 34 years as an employee of CAE Electronics Ltd.

The Early Trainers

The first approach to flight simulation was to use a real aircraft anchored to the ground. An example of this is the Sanders Teacher (Figure 1), introduced in 1910⁽³⁾. It consisted of a modified aircraft mounted on a universal joint attached to the ground and located in an exposed position so that it could face into the prevailing wind. It relied on wind and wind gusts to allow the trainee to practice moving the controls to test the response of the aircraft.

The Walters machine⁽²⁾ constructed in 1910, did not rely on the wind, but on instructors to create the disturbances while the student attempted to maintain equilibrium through the use of controls connected through wires and pulleys to the base.

A similar device was the Antoinette "apprenticeship barrel" trainer⁽⁴⁾ which was constructed also around 1910. It consisted of two half-sections of a barrel mounted and moved manually by instructors to represent the pitch and roll of the aircraft. The trainee sat in the top section and tried to keep a reference bar aligned with the horizon.

None of these devices were satisfactory. The wind was not reliable and the instructors tended to get out-of-phase with the student, probably providing negative training.

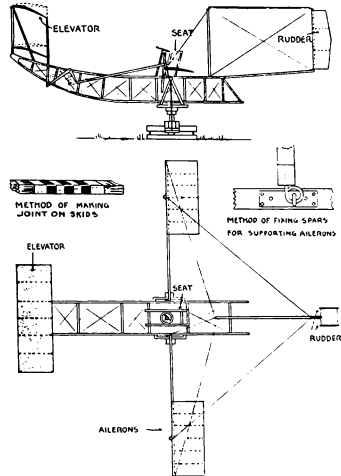


Figure 1. The Sanders Teacher

Courtesy of Flight International

The LINK Trainer

Various experiments continued in the 1920's using compressed air actuators or electric motors, instead of wind to produce motion^(12,14). They tried to produce the "feel" of the aircraft, but most failed because they lacked realistic dynamic responses.

The first effective ground based pilot training device did not appear until 1929 when a young aviator by the name of Ed Link designed the now famous LINK TRAINER⁽¹⁾ (Figure 2).

Edwin Link gained his early engineering experience with his father's firm, the Link Piano and Organ Company of Binghamton, New York. The trainer was developed in the basement of the Link factory and had actuators powered by air generated from organ bellows to move the trainer in pitch, roll and yaw. His idea was that a student

*Director, Research & Development
Member AIAA

could become accustomed to the aircraft controls and learn approximately what would happen in an aircraft as the controls were moved. Reference 1 states: "The whole unit sat on a universal joint and was moved by a series of bellows and motors that would gasp and wheeze as it whirled around to give the effect of an airplane in motion."

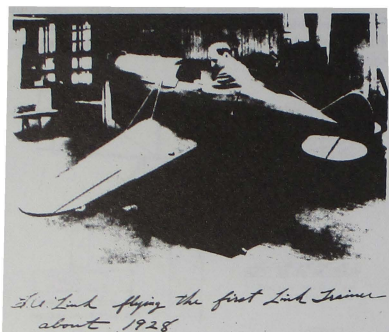


Figure 2. Ed Link in the first LINK TRAINER.

The first advertisement for the pilot-maker, which appeared in 1929, listed the Link Aviation Trainer as "an efficient aeronautical training aid - a novel, profitable amusement feature".⁽¹⁾

Ed Link's trainer was not initially accepted by the aviation world, but he was able to sell it to amusement park operators and a mechanism was installed on it to accept coins.

However, Ed was convinced that the device could be used to train pilots and started his own school, featuring the "pilot-maker" as a fundamental part. It was located in the basement of his father's organ factory and was called the Link Flying School. For eighty-five dollars he guaranteed to teach anyone to fly - this included ground school training plus two hours of flight time in an aircraft. In 1930 over one hundred persons soloed at the school, but then, with the depression, business declined.

In 1934 the US Post Office engaged the Army Air Corps to carry the mail. Unfortunately the pilots of that day had little experience in night and instrument flying and five airplanes crashed in the first few days.⁽¹⁾ In desperation, the Army acquired six LINK Model "A" trainers in June 1934 (Figure 3), and this marked the real beginning of the flight simulation industry. The trainers were fitted with instruments, operated either mechanically or pneumatically and since the trainer was able to rotate through 360 degrees, a normal magnetic compass was installed.

Ed Link continued to improve his device, adding more instruments, switches and even a radio compass and radio beacons. The trainer was further improved by attaching a course plotter to it. This consisted of a map on a table, over which a self propelled steerable "crab like" device crawled, leaving an ink trail of the simulated flight. The instructor could then check the student's position, and manually control the simulated radio beacon signals to the trainer. Hence it became a useful navigation training aid, although it is said that if the student got too far off course, the crab fell off the table and the exercise had to be started over again.

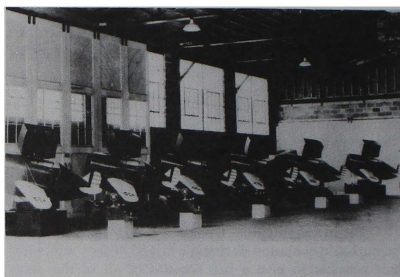


Figure 3. Six trainers for U.S. Army Air Corps 1934. First sale for pilot training.

Sales continued to increase throughout the 30's and by the time the U.S. entered the war in December 1941, thirty-five countries, including Japan, the USSR, France and Germany, were using Link-Trainers to train their airmen for war. The trainers were called "Blue Boxes" because they were always painted blue. Link sold over 10,000 of these trainers. More than half a million Allied airmen were trained in them during World War II and since that time Link's name has become synonymous with Flight Simulation.

The Celestial Navigation Trainer

The success of the Link instrument flying trainer prompted development of other training devices. One was the Celestial Navigation Trainer⁽¹²⁾ (CNT). In 1939 Britain requested Link to design a trainer to improve the celestial navigation skills of crews who were ferrying "surplus" US aircraft across the Atlantic to England. Such a trainer could also be used to improve the bombing accuracy of night raids over Europe. Ed Link, together with aerial navigation expert, P. Weems, designed a massive trainer which was capable of training an entire bomber crew.

The first CNT was delivered in 1941. The huge silo-shaped building contained a fuselage which could accommodate a pilot, navigator and bombardier (figure 4).

On a dome above the crew the major constellations, along with some 350 other stars, were accurately positioned.⁽⁷⁾ The movements of the stars were synchronized with the passage of time and aircraft position, permitting navigators to shoot fixes, during the simulated flight. Images of the terrain were projected on to a movie screen ahead of and below the trainer to give the navigator practice in picking out landmarks during simulated daylight flight. In addition to being a navigation trainer this device also trained the crew to work as a team. More than five hundred of these huge trainers were produced and put into operation in England and the United States.⁽⁸⁾

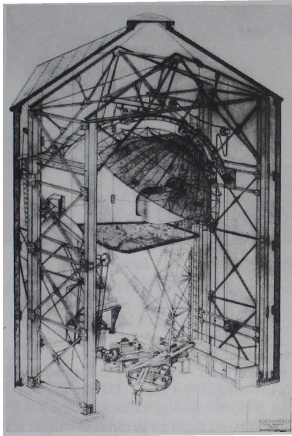


Figure 4. Link Celestial Navigation Trainer.

The Beginnings of the Analog Simulators

In the late 1930's in Britain, interest grew in developing a trainer, more sophisticated than the Link Blue Box. Aircraft themselves were becoming more complex with the emergence of the Hurricane, Spitfire and others. The trainee pilot could no longer quickly master all the aircraft systems and needed more formal training than had been required for the simple biplanes. Mock-ups of specific aircraft types were required.

One such family was the Silloth Trainers⁽⁹⁾. These trainers attempted to simulate all of the aircraft systems and their interactions. Their creator, Gordon Isles, interesting enough, had a background similar to that of Ed Link, in that his father worked for the Aeolian Pianola Company in Great Britain⁽⁷⁾ and he had a strong interest in pianola technology. After university he joined the Auxiliary Air Corps and became Chief Flying Instructor at Squires Gate, Blackpool. He was then posted to Silloth, England, to try to create a suitable training device for the

Lockheed Hudson aircraft which were frequently damaged during take-off. Work started in 1940 on a trainer for the Hudson, and soon afterwards, for the Halifax aircraft. The computer was basically pneumatic, adapted from musical technology, with electric elements to open and close valves and bellows to operate levers.⁽⁷⁾ One advantage was that the pneumatic instruments (airspeed, altitude, etc.) could be driven directly by tapping off air pressure from the corresponding point in the computer. However the computation seems to have been empirical rather than based on a mathematical model.

These Silloth Trainers were manufactured by the Automatic Player Piano Actions Ltd. in Southall. The Americans became interested in the Silloth idea and the U.S. Navy had one built by the Mohler Organ Company in Maryland in 1941. However, the U.S. climate caused pieces to warp resulting in unpredictable performance and even in Britain the Silloth approach was discontinued due to it's poor reliability.

The Electromechanical Approach

The Silloth Trainer demonstrated that a more reliable approach was needed to simulate the aircraft systems.

In Britain in the late 1940's an electromechanical analog computer for the simulation of aircraft longitudinal dynamics was proposed by G.M. Hellings, of the Ministry of Supply.⁽⁶⁾ Non-linear functions were to be generated by shaped cams. A mechanical version of this device, the Day Landing Trainer, was manufactured by General Aircraft Limited and used at the Empire Central Flying School. Further development of this device was carried out after the war at Air Trainers Limited of Aylesbury⁽¹¹⁾.

Air Trainers Ltd delivered an electromechanical Viscount 724 aircraft simulator to Air Canada in 1957. This was a most ingenious device consisting of a large array of small electric motors, gears, pulleys, levers, cams and switches (Figure 5). The interfacing of information from the mechanical computer to the cockpit was by means of long stainless steel ribbons running underneath the floor and up into the flight compartment to drive the flight instruments and indicators and to feedback the position of the pilot's controls, etc.

When Air Canada no longer needed this trainer, they gave it to the Central Technical School in Toronto. At the time of writing, it is still being used as a basic flight trainer for new students entering the flying school. This is a testimonial to the reliability of the mechanical computer.

The Electronic Simulator

Successful as it was, the Link Blue Box had shortcomings. The effects of the ailerons, elevators and rudder were independent, without any cross coupling as in the real aircraft. It did not contain a mathematical model of the aircraft and did not solve the equations of motion.

Improvements were continually being sought. As early as 1936 Mueller, at MIT, suggested that an electronic analog computer could be used to simulate aircraft longitudinal dynamics.⁽¹⁰⁾ In 1941 an electronic simulator which solved the aircraft equations of motion was designed at the Telecommunications Research Establishment (TRE) in Britain⁽³⁾, for use in a radar trainer. Although many different designs were created, the TRE group was disbanded after the war. The electronic simulation technology of the USA appears to have dominated post war development.⁽⁷⁾

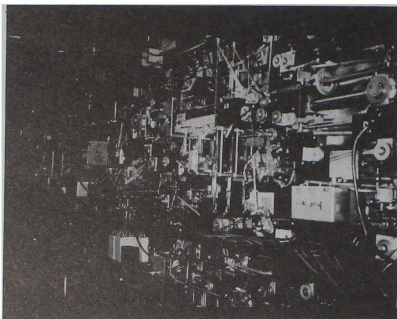


Figure 5. Viscount 724, Electromechanical Computer.
Note: Steel Tapes for Communication.

In the USA, as a result of the poor reliability of the Silloth trainer built by the Mohler Organ company, it was decided to build an electrical version. This was undertaken by the Bell Telephone Labs, who completed an operational flight trainer for the Navy's PBM-3 aircraft in 1943.⁽⁶⁾ It consisted of a cockpit with controls and instrumentation, plus an electronic computing device to solve the flight equations. Non-linear functions were generated by contoured potentiometers driven by integrator servos.

This trainer made use of work done by TRE in Britain, but is claimed to be the first electronic flight trainer to simulate the aerodynamic characteristics of a specific aircraft. The simulator had no motion, visual or variable control loading. A total of 32 of these simulators for seven types of aircraft were built by Bell and Western Electric.⁽⁶⁾

At about the same time Dr. Richard C. Dehmel originally an engineer with Bell Telephone, joined Curtiss-Wright and independently developed an electronic flight trainer for the B50 Bomber⁽⁹⁾. The first one was delivered to the US Air Force in 1950.

The Link company also worked hard to develop an electronic flight trainer and a strong competition developed between Link and Curtiss-Wright to produce an F-80 simulator for the US Air Force. The competition

was won by Link in 1949 and the trainer, designated the C-11, or Linktronic⁽¹⁾ was the first ground based trainer for a jet-powered aircraft (Figure 6). Ed Link reluctantly gave up the idea of a moving cockpit and for some years flight trainers were built without motion or visual systems. However, the C-11 included sophisticated flight instruments, engine and navigation simulation, including simulated malfunctions. Link eventually sold more than a thousand of these C-11 trainers.

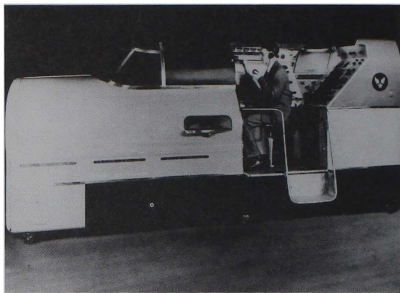


Figure 6. Link "C-11", First Jet-powered Aircraft Simulator (over 1000 built).

The Modern Simulator

It was in the early 1950's that the modern simulator started to take form. The early contenders in the field were Curtiss-Wright and Link. During the war, Redifon developed radio navigation trainers for the RAF and this experience allowed them to get into the flight simulation business after the war ended in 1945. Canadian Aviation Electronics (CAE), came into existence in 1947, repairing aircraft avionics equipment and after several unsuccessful attempts at building electronic consumer goods such as television sets, began building flight simulators under license to both Curtiss-Wright and Link. Link merged with the General Precision Equipment Corporation in 1954 and became known as the Singer-Link Company.

These early simulators were elementary by today's standards but they contained all the basic systems except motion and visual. The computations were analog, by means of amplifiers, computing servos and relays. Non-linear functions of variables were represented by wire wound contoured potentiometers (Figure 7). The shape of the potentiometer card varied the length of resistance wire per unit of brush travel and hence varied the voltage to generate the function. A number of functions of the same variable could be stacked on the same servo by having different diameter cards one inside the other. It was not very easy to change a function as this meant manufacturing and winding a new contoured potentiometer card.

An improvement to the AC analog devices was a switch to DC analog computing in the late 1950's. This

had a number of advantages. It allowed greater accuracy and more complex models. Functions were still produced by potentiometers and servos but they used taps and resistors instead of physically contouring the pots. It also allowed the automatic checking of amplifiers to be done by measuring the electronic tube grid voltage, thus assisting simulator maintenance. Everyone who worked on these machines had to know something about electronics. The computing circuits had to be adjusted frequently as the DC amplifiers used tubes which caused the output to drift.

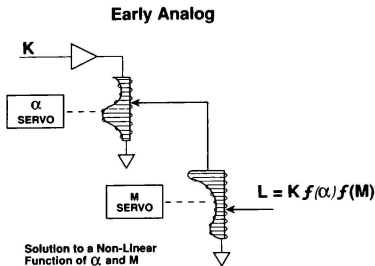


Figure 7. Wire Wound Contoured Potentiometers solve flight equation.

The Digital Computer

The idea of using digital techniques for flight simulation goes back as far as 1943 when the US Navy gave a research contract to the Servo mechanisms Lab at MIT to develop an Airplane Stability and Control Analyzer (ASCA).⁽²⁾ The idea was to have a universal flight simulator machine which could be used for both aircraft development and training. The machine developed for this purpose became the Whirlwind computer, but was never actually used for flight simulation. The University of Pennsylvania started the Universal Digital Operations Flight Trainer (UDOFT) project in 1950. The design was first published in 1954⁽⁴⁶⁾ and manufactured by Sylvania in 1960, proving that the aircraft equations of motion could be solved digitally.

In the early 1960's, Singer-Link started working on a special purpose digital computer for simulator use. After experimenting with the Mark I and Mark II, they designed and built the GP-4 special purpose computer which was first used in a DC-9 simulator.⁽¹⁾ This computer used a magnetic drum memory to store programs and a magnetic core memory to store data.

CAE switched to digital flight simulation in 1965 with the delivery to SWISSAIR and IBERIA of a DC9-10 simulator using an SDS-930 general purpose computer. The computer had only 40K of memory and yet could hold all of the simulation models and data and run them at 10 iterations per second.

The introduction of the digital computer gave a tremendous boost to the simulation industry. The restrictions imposed by the analog simulation techniques were removed and it became possible to implement complicated mathematical models, including second order effects. Navigational facilities such as radio stations which used to be limited to 12 or 15 at one time and had to be laboriously set up by the instructor, could now be increased to 500 or 1000 stations all available for use at any time and fully automatic and accurate.

Computer power has grown tremendously over the years allowing improvements in every area of the simulation including instructor areas, special effects, visuals, etc. In a modern simulator it is difficult to even quote the effective computer power as distributed processing including special processors, micro processors, etc. are used throughout the simulator complex.

Motion Systems

Motion systems have undergone a substantial evolution before reaching the mature design which exists today. The early trainers relied on motion as a fundamental feedback of pitch, roll and yaw position and Ed Link believed that motion was absolutely indispensable to the trainer.⁽¹⁾ However, after World War II the USAF and US Navy argued that motion in the Link "Blue Box" did not produce the feel of a real airplane and should be done away with. Even today there is controversy in the US military about the value of motion systems, especially for fighter type aircraft simulators.

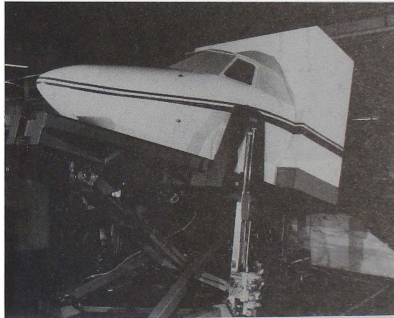


Figure 8. Link 3 DOF Motion System.

Motion systems started coming back into use in the late 1950's, with the production of 2 degree of freedom (DOF), 3 DOF and 4 DOF, both interdependent and independent systems. Link developed a 3 DOF system in the early 1960's, which became a standard in the industry for a number of years (Figure 8). Link also developed a 5 DOF cascaded motion in the mid 1960's and sold 96 to the US Army for UH1 helicopter simulators. Redifusion also built a number of different motion systems including

a large suspended 6 DOF system⁽¹⁴⁾. However, these designs were abandoned later in favour of the standard 6 DOF Stewart type system.

The first CAE synergistic 6 DOF motion system was built about 1969 for several DC10 simulators for KLM, SWR and UTA. There was a great concern among these airlines in those days about what would happen if a motion actuator were to break. In order for pilots to feel safe, CAE and other manufacturers had to add a second set of 6 actuators, called "safety actuators" just inside the first set. So these early 6 DOF simulators were driven by 12 actuators (Figure 9). As the reliability of the system was established, the inner actuators were replaced by large shock absorbers and crash pads and eventually deleted altogether.

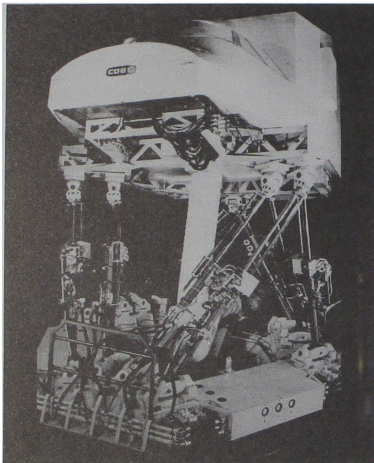


Figure 9. Early CAE 6 DOF Motion with Safety Actuators.

The next major advance in motion system design came in 1979 with the introduction of hydrostatic actuators. This design reduced the actuator friction to almost zero and eliminated the disturbing turn around bump, present on older motion systems. With the introduction of full digital control in 1984 a mature design has emerged which is essentially standard throughout the industry.

Visual Systems

Systems for producing "out-the-window" visual scenes have been proposed and constructed for almost as long as flight trainers have existed. However, realistic and flexible visual attachments have appeared only recently. A very large effort has been devoted to visual attachments, since one of the most important areas of pilot training has

always been visual take-offs and landings, especially under adverse conditions. In spite of the current trend towards the automation of transport aircraft, manual flight training is still a requirement and in the military there is a need for sophisticated visual systems to simulate battle field environments.

Numerous approaches to the generation of visual scenes have been attempted, many of which although unsuccessful, still advanced the state-of-the-art. The point-light source projection, or shadowgraph, method was popular in the 1950's especially for helicopter simulators where ground velocity was low. The CAE F104 simulator, built in the early 1960's used a point-light source to project a flat gridded earth onto a screen. From there it was viewed by a closed circuit TV camera and displayed to the pilot. The grid moved with aircraft speed and heading and had the correct perspective. It produced a good horizon and gave good pitch and roll cues. However, the picture was very unrealistic and although the grid changed size with altitude, it could not really be used to judge height for landing. Still, some 32 of these F104 simulators with visuals were built and sold to various NATO countries in 1965/66.



Figure 10. CAE Model Board Visual for a CH-47 simulator for Iran, showing mostly desert.

An improvement on this approach is the model board visual and a number of these were built by CAE, LINK, Redifusion and others. Some are still in use today. The model board consists of a three-dimensional scale model of the area to be simulated (Figure 10). A closed circuit TV camera is servo driven to "fly" over the model and relay the picture back to the pilot. The camera has special optics to allow it to get very close to the model.

This approach gives an excellent picture with a lot of detail especially close to the ground where it is important for landing (Figure 11). However, the model board has disadvantages. The pilot is restricted to a small flying area and the model itself is large and expensive. There is also the danger that the pilot may fly too close to the model and damage it.



Figure 11. Pilot's view of Mosque from Fig. 10 visual.

Other approaches have been tried. One was the LINK VAMP, system where a wide 70 mm movie film was taken along an ideal approach to the airport. This, when projected, gave a good quality image. The speed could be varied, within reason, and the wide film allowed some deviation in position away from the specified track. However, the pilot was still limited to a narrow corridor and had to fly along a predetermined route.

The real breakthrough in visual simulation came with the digital computer. The development of high density integrated circuits in the early 1980's, provided the intensive computing power necessary to generate realistic daylight visual scenes. The quality of visual image generation and display improved dramatically during the 1980's to the extent that zero-flight-time training is now possible.

However, the first computer generated terrain visual appeared in the early 1960's when, the General Electric Company (USA), as part of their space simulation effort, demonstrated a computer generated, repetitive patchwork terrain model.⁽²⁾

The first successful Computer Generated Image (CGI) systems for commercial simulators, were for night scenes only. In 1969, McDonnell Douglas came out with their Visual Takeoff and Landing (VITAL) II visual system. Here, strings of light points were modelled to represent the runway centerline and edge lights and were presented at the correct perspective to allow a pilot to practice night time take-offs and landings. The traditional raster scan CRT display was replaced with the calligraphic technique, which gave a much sharper reproduction of light points.

Runways markings and texture were added in upgrading to the VITAL III, the picture was further improved by adding buildings and landscaping to produce the VITAL IV and real daylight capability added to produce the VITAL V, VI and VII.

Many companies now produce CGI visual systems. The high end includes Redifusion's SPX WIDE, CAF's MAXVUE, LINK MILES with their IMAGE series, GE's COMPU SCENE series and Evans & Sutherland with their ESIG series. There are many other companies producing lower cost visual image generators, and as the quality improves, some of these will create serious competition to the high end systems.

Continuous improvement in computing power and modelling techniques have allowed tremendous improvements in visual scene detail, colour and shading in recent years. Pilots are able to recognize and practice landing at all airports along the airline company routes. A variety of special effects can be generated realistically, including clouds, thunderstorms, rain, snow, lightning and air and ground hazards such as other aircraft and ground traffic on runways.

Special Visual Displays

One limitation to all the visual systems discussed so far is the field of view. Even the so called WIDE displays cannot exceed a field of view of about 40° vertical by 200° horizontal. This is fine for commercial simulators where the main interest is in teaching the pilot how to approach and land at an airport. However, for military simulators where the requirement is to train pilots in air-to-air or air-to-ground combat, this field of view is completely inadequate.

The Dome Display

One solution to this is the DOME display where a large dome is placed over the entire simulator to form the projection screen. This idea is not new. The shadowgraph visual on the Shorts Helicopter Simulator, described in reference 6, is essentially a dome visual produced in 1955.

In the last two decades many other companies, such as LINK with their ESPRIT, McDonnell Douglas at St. Louis and Mesa, Hughes Aircraft, G.E., E&S, British Aerospace at Warton, Thomson-CSF in France and others, have experimented with this type of display. The domes are mainly sold to the military for use with fixed base, fighter type aircraft simulators. Although the picture quality, especially with head tracking can be good, the domes are large, expensive and not suitable for use with motion systems.

The Helmet Mounted Display

Another solution is the Fibre Optic Helmet Mounted Display (FOHMD), such as the one developed at CAE Electronics in Montreal. This is an area-of-interest approach where the display is attached directly to the

pilot's helmet. The pilot's head and eye positions are monitored and used to control the computer generated image. The image is transmitted from high power projectors to the helmet through high-resolution coherent fibre optic cables. Because the head and eye positions are tracked, the pilot's field of view is unlimited. Since the image needs to be generated only in the relatively small area where the pilot is looking, the demands on the image generator are low.

The FOHMD is favoured by the military who need the performance it offers and are accustomed to wearing helmets. As the technology advances, lower cost, lower performance versions are appearing with miniature CRTs located directly on the helmet, eliminating the large projectors and fibre optic cables. These new systems have potential for other applications such as telepresence, virtual reality, etc.

Data Problems

As long as analog computers were used, there was not much emphasis on obtaining accurate data from the aircraft manufacturers on the performance of their airframes and engines. Simulator performance was usually adjusted empirically according to the pilots comments. Furthermore, the analog computer did not facilitate complex or highly accurate models. A large number of second order effects had to be avoided, as every input to a summing amplifier added noise, even if the input was zero. Too many inputs could degrade the computing accuracy.

However, with the coming of the digital computer, plus the FAA Advanced Simulation Plan, the need arose for much better aerodynamic and system data.

The only flight test data generally available, was that required for aircraft certification, which tended to be taken under extreme weight and C.G. conditions in order to prove aircraft performance limits. Aircraft performance data for simulation was needed well inside the flight envelope at mid C.G. and weight conditions.

In order to improve the situation it was necessary, first of all, to define precisely what the simulator data requirements were. Therefore, in 1976, the commercial flight simulation industry, represented by various airlines and all major aircraft as well as simulator manufacturers, met in Montreal under the auspices of the International Air Transport Association (IATA). At this meeting the IATA Flight Simulator Data Working Group was formed, with a mandate to establish a detailed specification of simulator design and validation data.

After many meetings the Working Group issued the first edition of the "IATA Flight Simulator Design and Performance Data Requirements" in October 1980, specifying data required to meet the highest level of simulator fidelity, i.e. for FAA Phase II and Phase III approvals. This document has been revised several times with the 3rd edition being issued in 1990.

The Boeing Aircraft Co., always a forward looking organization, set up a special department to generate simulator data. Of course this service was not free, and the data package became a substantial percentage of the total simulator cost.

However, in spite of the publication of the IATA document and the financial rewards for supplying data, many other aircraft manufacturers were reluctant to meet the IATA standards, as it required additional work and often additional flight testing. Fortunately, in recent years most manufacturers of large aircraft, realize that the simulator data package has to be part of aircraft development and include the necessary tests in the design process. The quantity of aero data has grown exponentially over the years, as illustrated in Figure 12.

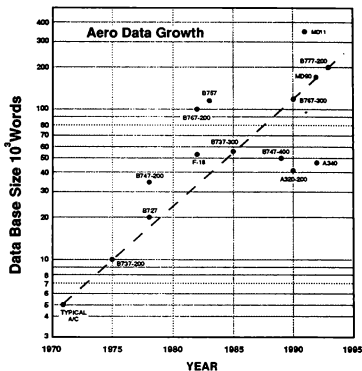


Figure 12. The Size of Simulator Aero Data Bases

However, for small or older aircraft, there is often no data package available. This has given rise to special data gathering companies such as Kohlman Associates in USA, NRC in Canada, DLR in Germany, etc., who for a suitable fee will instrument and fly the aircraft and collect and analyze the data. On some occasions the simulator manufacturer together with a research organization may collect and analyze the data, as for example CAE with Delft University for the Citation 500 in 1986 and with NRC for the Dash 8 in 1990.

Programming Languages

It is interesting to note the evolution of the mechanization of mathematical models into the simulator. In the days of analog simulation, the modeller had to be able to design the various electronic circuits necessary to mechanize the simulation models. The first digital computers which appeared in the mid 60's were programmed in assembler language and sometimes even in machine language. Equations had to be scaled and a good knowledge of how the computer processed the data was required in order to produce an effective model.

With the introduction of Sigma 5 and DEC VAX computers in the 1970's, it became possible to produce real time programs in Fortran. This made life much easier, as scaling and memory addressing were no longer a problem, and equations could be written in a normal way. However, some knowledge of programming and the storage of data was still required. (More recently the "C" language, which came out of the Bell Labs, has become popular, especially for non-mathematical programs). In 1979 the US Dept of Defense (DOD) set up a competition for a standard language for military use. The competition was won by Ada (named after the niece of the English poet Lord Byron) and was first issued in 1983 as DOD standard 1815 (year of Ada's birth). It has been revised several times, the latest issue being Ada 9x. This program allows a lot of flexibility in programming, but it is complex and difficult to learn. It is used mainly in large US military real time programs, but has not been favourably received in the commercial simulator world, and has not become the universal simulator language as the US military had planned.

As computer programs get larger and more complex, the trend is toward object oriented software languages and Computer Aided Software Engineering (CASE) tools, which further removes the operator from the computer hardware. CAE has developed such a tool, called Realtime Object Oriented Software Environment (ROSE), specifically for simulation. CAE started using ROSE for nuclear power simulation in 1991, and it's use has been steadily growing since then.

With ROSE the operator graphically draws the system to be simulated on the CRT screen, by choosing objects or icons from a library and arranging them in a pictorial fashion. The operator doesn't have to have programming skills and doesn't even have to know in what language the ROSE utility will implement the code, which could be Fortran, C or Ada. Modifications to the model can be easily implemented by non-engineering personnel or by airline technicians, by updating the graphical display on the CRT screen.

Regulation of Simulators

It seems appropriate in this paper to discuss also, the evolution of the regulations governing the approval of flight simulators for training, as these regulations have closely paralleled and influenced the development of the simulators themselves.

When simulators started to be used to train the pilots of passenger aircraft, it became apparent that certain standards and some form of regulation was required. Although most countries eventually created a regulatory authority, the Federal Aviation Administration (FAA) of the USA was the leader and most other regulations have been based on the FAA.

The FAA has been involved in simulator evaluation and approval for four decades⁽⁹⁾. Early credits for the use of simulators were for IFR flight only since no useful visual system was available.

However, as early as 1954, air carriers were allowed to perform all but four proficiency check manoeuvres in a simulator. These four were: engine-out takeoff, engine-out landing, ILS approach and missed approach. Rapid advances, during the 1960's and early 1970's in computer and visual technology, allowed expansion of simulator usage and consequently required the development of standards defining the fidelity necessary for modern simulation. For a full account the reader is referred to reference 8.

The earliest FAA specific regulations for the approval of Flight simulators was FAR Part 121, Appendix B, January 1965, "Minimum Standards for the Approval of Airplane Simulators".

In 1969, the FAA issued a notice of proposed rule making to provide more extensive use of the flight simulator. This, after consultation with industry, resulted in publication in December 1969 of Advisory Circular AC121-14 "Aircraft Simulator Evaluation and Approval". This was updated early in 1976 to become AC121-14A; however, training manoeuvres involving ground effects or ground handling still had to be done in the airplane.

In May 1977 UAL asked the FAA for approval to do take-off and landing checks in the simulator for both recency of experience and transition training. They instrumented a B727 and a DC10 aircraft to gather additional flight test data for this purpose and were given "Landing Manoeuvre Approval" for these aircraft.

In 1977, the FAA considered for the first time an "Advanced Simulation" concept in which, ultimately all training and checking could be done in a simulator, provided the technology met certain minimum standards.

In late 1978, AC121-14A was replaced by AC121-14B, defining Simulator Visual System requirements for the first time and also allowing Landing Manoeuvre Approval (LMA) for pilot recency of experience qualification.

Because of the success of the LMA program and continuing improvements in simulation technology the FAA proposed the "Advanced Simulation Plan" in 1979. A new Appendix H to FAR 121 was adopted in June 1980, specifying the simulator requirements for achieving Phase I, II and III levels of approval. Higher phases indicated higher levels of sophistication and permitted more training and checking. Appendix H permitted "total" or zero flight time (ZFT) simulation for transition and upgrade training with a Phase III simulator. A new advisory circular AC121-14C was issued at this time, to explain at least one method of meeting the requirements of Appendix H, plus requirements for existing non-visual and visual levels of simulation (not in Appendix H).

In 1982 organizational changes were made in the FAA and a renumbering of the advisory circulars. Hence AC121-14C was re-issued as AC 120-40 in January 1983. Additional clarification, mainly in the visual area, was added and the document became AC120-40A in July 1986.

In 1987 it was decided to consult with industry in order to review the performance validation parameters. A working group was set up with representatives from airlines, simulator manufacturers, ATA and government and after much discussion the document was completely rewritten and finally issued as AC120-40B on 7 July, 1991. Levels of simulator sophistication were designated as A, B, C and D instead of a Phase I, II, etc.

International Standards

As flight simulators started to come into general use in the 1960's and 70's, most countries set up their own regulatory authorities, many patterned on the FAA, but with their own special requirements.

As more simulators came into use, airlines found that they could rent out surplus time to other airlines. It became a profitable business, often paying off the capital cost of the simulator in a short time. However, every airline renting simulator time abroad, had to have the simulator qualified by their own regulatory authorities. This added an unwelcome burden to all concerned.

In an effort to improve this situation, the Royal Aeronautical Society's Flight Simulation Group convened meetings in September 1989, in London and in February 1990, in Long Beach, California, to discuss the possibility of an international standard. These meetings were attended by over 350 delegates representing regulatory authorities and industries from all major world countries including the Soviet Union. It was decided to form a working group to try to formulate a set of standards that could be accepted by most countries and industries. Although this sounded like an impossible task, the working group succeeded, after many meetings in many countries and using the FAA AC120-40B document as a baseline, in producing a document entitled "International Standards for the Qualification of Airplane Flight Simulators". Only two levels of qualification are specified corresponding to FAA AC120-40B level's C and D. A draft of this document was presented and approved by a Royal Aeronautical Society Conference in London, in January 1992.

It was then submitted and approved by the International Civil Aviation Organization (ICAO) as the international standard in February 1993. It will now be translated into the other four official languages (French, German, Arabic and Spanish) and released later this year. The new regulations are expected to be ratified by each separate government in the near future.

Major Turning Points

An interesting study of the simulation industry was done recently by Drs. Miller, Hobday and others at the University of Quebec in Montreal.⁽⁴⁾ They found that the development of flight simulators, unlike that of other manufactured products, depends on a close working relationship between manufacturers (simulator and aircraft), customers (airlines) and regulatory authorities.

A strong interaction between these groups in recent years, has had a major influence in the development of the modern flight simulator. Since 1950 the following turning points have had a major impact on the growth of the flight simulation industry.⁽⁴⁾

1950-1960	Kick Off Orders (Airlines "kick-started" the industry)
1965	Shift to Digital Computing
1970	Introduction of High Performance Motion Systems
1978-1980	FAA Advanced Simulation Plan
1981	CGI Visuals
1983	Emergence of Complex Electronic Aircraft
1984	Deregulation of Airline Industry
1990	Popularity of FTDs
1990	Advanced Qualification Program (AQP)

Many of these issues have already been addressed. Some additional points are discussed below.

Kick-Off Orders

In the late 1950's several of the world's leading airlines such as BOAC, KLM, SWR and PAN AM foresaw the advantages of simulator training and invested in the future by ordering Full Flight Simulators (FFS). This "kick-started" an industry which up to then had relied on sporadic military orders.

Since the airlines were investing in their future, they wanted a close working relationship with the manufacturer. Many of the advances in simulation were made jointly by the simulator manufacturer and the airline pilots. In contrast to the military procurement, an atmosphere of trust developed between airlines like KLM, SWR, BOAC, etc. and the supplier (at least this was true with CAE and I expect with other manufacturers as well). This allowed contracts to be flexible and changes that were mutually beneficial could be made during the design process. CAE strongly believed in progress through evolution and engineers were given a free hand to make continual improvements to the product. The final judge was, of course, the customer pilot who was not afraid to express his opinion. I believe this process contributed greatly to the evolution of the modern flight simulator.

Intelligent Aircraft

The appearance of the glass cockpit, and fly-by-wire aircraft with complex avionic "black boxes" in the early 1980's increased the cost and complexity of simulators and presented a new challenge to simulator manufacturers.

New technical issues arose, such as whether to purchase the expensive black boxes and stimulate them, or attempt to get sufficient data to simulate them. The problem with stimulation was to produce "simulation functions" such as repositions and freezes. This led to intensive discussions between the users and the avionics manufacturers such as Honeywell, Collins, Sperry and others. A working group was formed, and a new specification, ARINC 610, issued in 1985. This specified a number of SIMSOFT functions, such as repositions, locks and freezes, which should be included in the avionic black box design so they could be used in simulators as well as aircraft. This document is presently being updated to make it more comprehensive and is expected to be released later in 1993 as ARINC 610A. Of course great care must be taken not to compromise aircraft safety. All new aircraft such as the B777 as well as the A320/A340 etc, will have boxes built to these standards. This is probably the first time that simulator requirements have caused a change in the aircraft design. We can expect to see more such developments in the future as the aircraft and simulator design begin to merge. After all, the technology involved in fly-by-wire aircraft systems, or in the B777 Airplane Information Management Systems (AIMS), is similar technology to that used in the design of flight simulators.

Deregulation

The deregulation of the airline industry in the early 1980's increased competition and placed more emphasis on simulator costs. This started a trend toward greater use of lower cost training equipment, such as Flight Training Devices (FTDs) with limited simulation and Computer Based Trainers (CBTs) which can provide effective teaching of subsystem fundamentals. In order to provide some control and to allow credits to be obtained with various levels of FTDs, the FAA issued new regulations, AC120-45 in May 1987 and the more comprehensive AC120-45A in May 1992, to cover these devices. These events have given a boost to the manufacture of lower cost devices and have allowed smaller companies to compete with the large simulator manufacturers.

AQP

The Advance Qualification Program (AQP) regulations, AC120-54 were first issued in August 1991 and programs are now being developed by simulator manufacturers and airlines. In essence, the AQP evaluates the effectiveness of the training program, rather than just the performance of equipment used. For example, the interaction of the flight crew, i.e. Crew Resource Management (CRM), is given particular attention in the training scenario. This emphasis is due to the fact that a number of aircraft accidents have been attributed to poor communication among the airline crew.

Under the AQP, more emphasis will be on human factors and crew performance monitoring, resulting in use of a wider range of low cost flight training devices.

CONCLUSIONS

Flight simulation technology has come a long way since the days of the famous LINK blue box. Powerful digital computers now make "zero flight time" training a possibility.

In such a rapidly changing industry it is difficult to try to predict even five years into the future. However as computers become more powerful and less expensive and as airlines are having to cut costs there will be pressure to make greater use of lower cost devices such as FTDs, CBTs, etc. Higher level languages such as ROSE and more automated checkout procedures will come into common use.

The technology developed for flight simulation has already been used in powerplant simulators and no doubt will find uses in many other areas such as medicine, mining, entertainment and even in the design of aircraft themselves.

REFERENCES

1. Kelly, L.L., as told to Parke, R.B. (1970) *The Pilot Maker*, published by Grosset & Dunlap, New York
2. Rolfe, J.M., Staples, K.J., *Flight Simulation Chapter 2, "A short history of the flight simulator"*, published by Cambridge University Press, England, 1986.
3. Heintzman, R.J., "Flight Simulation: History, Evolution, and Future Directions", Aeronautical Systems Division, United States Air Force.
4. Miller, R., Hobday, M., "Innovation in High-Technology Craft Industries: The Case of Flight Simulation", University of Quebec in Montreal report, 1992
5. Haward, D.M., "The Sanders Teacher", *Flight* 10 December 1910
6. Adorian, P., Staynes, W. and Bolton, M. (1979). The Evolution of the Flight Simulator. In "Fifty Years of Flight Simulation", London: Royal Aeronautical Society.
7. Rolfe, J.M., Bolton, M., *Flight Simulation in the Royal Air Force in the Second World War*, Paper No. 1560, The Aeronautical Journal of the Royal Aero Society, 1988.
8. Booth, E., The Evolution of FAA Guidance for Simulator Qualification. Proceedings of the Sept 1989 Symposium of the Royal Aeronautical Society.
9. Federal Aviation Regulation (FAR) Part 121. Appendix H, August 1980.
10. Mueller, R.K., MIT "An Electrical Device for Solving the Equations of Longitudinal Stability", Journal of the Aeronautical Sciences, March 1936.
11. Air Trainers, Ltd., "In on the Ground Floor - Twin Simulator with Mechanical Computer", *Flight* 23 April, 1954.
12. Killgore, J.I., "The Planes That Never Leave the ground", A History of Flight Simulation, published by LINK Flight Simulator Division.
13. Cutler, A.E., "Environmental Realism in Flight Simulators", The Radio and Electronic Engineer, Jan 1966.
14. Rolfe, J.M., "Keeping up on the ground", Aeronautical Journal 1977.
15. Ringham, G.B., Cutler, A.E., "Flight Simulators", Journal of the Royal Aero Society, March 1954. (Volume 58)
16. Dunn, W.H., Eldert, C., Levonian, P.V., "A Digital Computer for Use in an Operational Flight Trainer", JRE Transactions-Electronic Computers, June 1955.

INTERNALLY COHERENT SYSTEM OF INNOVATION: THE CASE OF FLIGHT SIMULATION

by
Roger Miller, D.Sc.
F. Xavier Olleros, Ph.D.
Hydro-Quebec Chair in Management of Technology

Abstract

This paper reports on the finding of a study of the flight simulation industry, from a perspective of industrial economics. Our analysis corroborates two basic insights from the recent literature on industrial dynamics: first, that cooperation and competition are complementary, rather than contrary, aspects of industrial evolution; second, that both cooperation and competition are necessary for the emergence and effective functioning of complex industries. Our main original contribution consists in suggesting that, under conditions of increasing returns to market share, tight oligopolies are ideally suited to deliver an optimal combination of inter-firm cooperation and competition and, hence, a consistently high level performance.

I. Introduction

Contemporary science derives much of its impressive thrust and coherence from its *anti-reductionism* and its *anti-determinism*. The former - encapsulated in the principle of emergence - stresses that the whole is qualitatively different from the sum of its constituent parts. The latter - captured by the principle of self-organization - tells us that no exogenous set of variables can completely determine the behavior of a complex system.

The social sciences in general, and industrial economics in particular, trail behind the natural sciences in the effort to loosen the deterministic-reductionist straightjacket [27; 30]. Recently, however, both the evolutionary and the institutional schools of economics seem to be heading in the right direction, that is in the direction of recognizing and trying to grapple with nonlinear causalities, path dependence, synergistic effects, endogenously structuring

processes, evolutionary pathologies, indeterminate outcomes, etc. [25; 4; 39; 15; 11; 10; 33; 16].

In short, the twin disciplines of industrial economics and strategic management are finally on their way toward being able to explain and exploit "complexity", in the strict, scientific sense of the term [42; 26].

In this article, we report on our findings from a study of a high-technology craft industry (cf. Miller, Hobday, Olleros and Leroux-Demers, 1993), namely the civil flight simulator industry. We intend to show that this is a truly *complex* industry, whose emergence and impressive performance can only be understood from a self-organizing perspective. We also hope to show that in this kind of industry, old dichotomies (e.g., chance vs necessity, choice vs determinism, efficiency vs market power, cooperation vs competition, ...) take on a totally new, and more constructive, light. Relatedly, we will argue here in favor of a less ideological and more nuanced evaluation of tight oligopolies than the one currently espoused by most industrial economists.

The article is structured in five parts. Following a short discussion of the literature on the determinants of industry performance, the central section of the article tries to explain the successful emergence and consistently high performance of the flight simulation industry. A discussion of some of the issues raised by our analysis is followed by a concluding section.

II. Explaining Industry Performance

Why is it that some industries consistently deliver ever more performing products - or high value services - at ever lower prices, while other industries struggle along with the same old mediocre offer, at prices that are perhaps rising rather than falling? What explains the performance differential of the

various industries, both nationally and internationally?

Leaving aside, for the moment, industry life cycle considerations, the conventional explanation of differences in industry performance is that of the "structuralists" [12; 36]. According to this perspective, a poor industry performance - as reflected in a high inefficiency, a slow technical progress, inflated prices, etc. - typically results from collusive arrangements among oligopolists, arrangements which in turn can be traced back to a combination of high levels of industrial concentration and high entry barriers. In this view, therefore, tight oligopolies are, by their very nature, suspect and deserve close scrutiny by anti-trust authorities. By the same token, in this view, an industry's performance will be all the better the more its structure approaches the competitive ideal.

In what follows, we will use the case of the flight simulation industry to show that an efficient oligopoly is far from being a contradiction in terms. We will also show that industry evolution is a self-organized, indeterminate process subject only to one major constraint, that of internal coherence [35].

**III. A High Performance Archetype:
Flight Simulation**

The flight simulation industry has performed extremely well, ever since it became established. Year after year, it has delivered a steady flow of increasingly complex, sophisticated and reliable products, at reasonable prices. The industry is evaluated positively not only by its main participants, but also by clients and regulators. As Table 1 indicates, the flight simulation field has yielded high levels of innovation, without locking itself on inferior technological paths.

RANK	
• SYSTEM HAS YIELDED HIGH LEVELS OF INNOVATION	1
• PROCESS IS LED BY A RELATIVELY STABLE CORE OF PARTICIPANTS	2
• NEW ENTRANTS ARE GIVEN OPPORTUNITIES TO PARTICIPATE	3

Table 1. Effectiveness of flight simulation field

Technical progress in this industry can be measured along many dimensions. Table 2 shows a manyfold increase in iteration rates, that is the computation speed affecting the time span between pilot actions and system responses. Yet, as Table 3 indicates, the prices of simulators in real terms have remained at about the same level as in the early 70s.

The paradox to explain is the following: how could the field of flight simulation perform so well in terms of innovation while facing high levels of uncertainty and complexity.

The traditional approaches to dealing with the uncertainties inherent to innovation are: (i) diversification of innovative efforts along diverse research avenues, (ii) *Ex post* selection by market forces of the various entrepreneurial solutions, (iii) alliances and networks. However, the flight simulation field has developed an enduring system of innovation characterized by internal coherence and high performance without using any of these approaches. How did such a system come about?

COMPONENTS OF FULL FLIGHT SIMULATOR	1965	1985	1992
•FLIGHT SYSTEMS	20 Hz	30 Hz	60 Hz
•VISUAL SYSTEM	10 Hz	30 Hz	60 Hz
•INSTRUMENT INSTRUCTOR	10 Hz	15 Hz	60 Hz
•STATION	2.5 Hz	15 Hz	30 Hz
•MOTION	5 Hz	15 Hz	60 Hz
•ENGINES	5 Hz	5 Hz	60 Hz
TYPICAL AIRPLANE	BOEING 707	BOEING 747-400	BOEING 767
INDEX OF COMPLEXITY	100	186	361

Source: Authors' estimates from diverse sources

Table 2. Iteration rates - improvement (1965-1992)

HIGH TECHNOLOGY CRAFT INDUSTRIES - A RESEARCH PROJECT IN PROCESS

The main purpose of the research carried out at the Hydro-Quebec Chair in the Management of Technology is to uncover the logic of innovation prevailing in high technology craft industries, such as aerospace, flight simulation, nuclear power, electric networks, etc. High technology craft industries are typically linked to the development and exploitation of large technical systems [22]. The distinguishing attributes of these industries are the following:

- *Transactions are infrequent* but of very high value. Products are complex and they entail high levels of R&D knowledge, multiple technologies and systems integration. Typically, products sell for many millions of dollars and require lengthy periods of information exchange between buyers and producers [24].
- *Concerns for safety and quality requirements are extremely high.* Inferior products cannot be simply screened out by the market but must by design excel along a number of attributes such as reliability, fidelity, safety, etc. Failures are not tolerable: reputations could be lost overnight because of occasional accidents. If accidents do happen their high visibility triggers public pressures to reduce the likelihood of further failures (Perrow, 1988).
- *Buyers are informed and demanding decision-makers* spending large amounts of time and money at every transaction. As a consequence, transactions are informationally dense. Typically, buyers are large organizations with complex technical systems and high levels of expertise. They therefore expect to have a considerable influence on the parameter and design choices of their suppliers [24].
- *Products are customized* to fit clients' specific requirements: standard designs rarely exist. Complex products are engineered through long-lasting interactions between buyers and sellers. For instance, the design and implementation of a power network control system spreads over a period of 10 years.

As indicated above, the aim of our research program is to come to understand the industrial dynamics of high technology craft industries, particularly in relation with the development and adoption of new technologies. The present paper reports on our initial study focussing on flight simulation systems.

As regards research strategy, we discarded cross-sectional hypothesis-testing surveys, as well as narrow case studies (Bijker & Hughes, 1987). Given the particular nature of complex technological systems, we felt that the most appropriate strategy would be to aim for "thick descriptions" containing a wealth of detailed information about the technical, social, economic and political aspects of relevance. "Middle range" concepts were used to create conceptual order and enhance understanding (Constant, 1980). A review of the pertinent literature suggested a number of research questions and issues for discussion with industry executives, with a view to understanding the peculiarities of the innovation process in the flight simulation industry. The data were collected in a structured sequence of steps.

- Two *panels* with industry experts were set up to orient the study: a first panel was held in London, U.K., to initiate the study. Later on, a workshop on the evolution of simulator technology was held in Montreal, Canada.

- A round of structured *interviews* was held to investigate the various research questions. Issues for discussion were sent in advance to senior officers in the following types of organizations:

• Air carriers	8
• Public regulators	5
• Simulation systems developers	14
• Sub-systems or components makers	15
• National or international industry associations	9
• Aircraft designers and builders	6
• Research and military institutes	9
• Flight training centers	4
Total	70

- Follow-up *questionnaires* were sent to the 70 executives who had been interviewed, to try to sharpen and quantify our findings. A distinct, complementary questionnaire was answered by 35 small firms involved in sub-systems and flight training devices. In total, 71 responses were collected for statistical analysis.

	PRODUCERS OF ENGINEERED PROTOTYPES		
	FULL FLIGHT SIMULATORS	FLIGHT TRAINING DEVICES	GENERIC TRAINING DEVICES/ COMPUTER-BASED TRAINING DEVICES
PRODUCT ATTRIBUTES			
NATURE	CUSTOM ENGINEERED PROTOTYPE	CUSTOMIZED AND SPECIFIC PRODUCTS	GENERIC PRODUCTS
PRICE	\$10 TO \$17 M	\$2 TO \$8 M	\$100,000 TO \$2,000,000
EXAMPLE OF AIRPLANE REPLICATED	AIRBUS 320, BOEING 747-400 MISSION FIGHTER	AIRBUS 320, BOEING 747-400 MISSION FIGHTER	PRIVATE JETS OR PLANES, PART TASK
USE OF PRODUCT	CHECKING OF PILOTS TRAINING FOR RECOVERY IN DIFFICULT FLIGHT CONDITIONS	LEARNING OF COMPLEX TASKS	- TRAINING FOR SPECIFIC TASKS - TRAINING IN GENERAL AVIATION
DEVELOPMENT OF SIMULATOR	SPECIFIC TO EACH AIRPLANE SERIES AND/OR AIRPLANE	SPECIFIC PRODUCT FOR PLANES OR TRAINING TASKS	GENERIC PRODUCT FOR PLANE AND TRAINING
CLIENT CHARACTERISTICS	<ul style="list-style-type: none"> - SPECIALISTS MANAGING TRAINING CENTERS (AIR CARRIERS OR MILITARY PROGRAM MANAGERS) - HIGH LEVELS OF TECHNICAL EXPERTISE 	<ul style="list-style-type: none"> - SPECIALISTS MANAGING TRAINING CENTERS (AIR CARRIERS OR MILITARY PROGRAM MANAGERS) - HIGH LEVELS OF TECHNICAL EXPERTISE 	<ul style="list-style-type: none"> - NON-EXPERTS WHO OFTEN LACK TECHNICAL EXPERTISE TO JUDGE PRODUCTS
CERTIFICATION BY REGULATOR	<ul style="list-style-type: none"> - REGULATORS INVOLVED IN ASSESSMENT TO CERTIFY SIMULATORS BY LEVELS - REGULATORS OBLIGE SUPPLIERS TO PROVE ORIGIN OF DATA 	<ul style="list-style-type: none"> - PRODUCTS SUBJECT TO LOWER FIDELITY REQUIREMENTS AND NORMS - SIMULATOR MUST MEET FIDELITY REQUIREMENTS OF PRODUCTS' CLASSIFICATION LEVELS 	<ul style="list-style-type: none"> - PRODUCTS ARE OFTEN NOT SUBJECT TO CERTIFICATION AND INSPECTION - NO OBLIGATION TO PROVE DATA ORIGIN

Table 4. Comparison of prototype and generic devices sub-groups

SIMULATORS	YEAR	CONSTANT PRICE
AIRBUS 300	1972	12.8 MILLION \$ U.S.
AIRBUS 300	1979	12.9 MILLION \$ U.S.
AIRBUS 310	1981	10.3 MILLION \$ U.S.
AIRBUS 300-600	1983	11.5 MILLION \$ U.S.
AIRBUS 320	1992	12.3 MILLION \$ U.S.

Source: Authors' estimate

Table 3. Price of simulators in real terms

a) *The social construction of a technical system*

Flight simulation was born when Ed Link, a builder of organs, patented a flight trainer in 1929. During WWII, simulators were built to help reduce training accidents. In 1951, Redifon (now Rediffusion) built a Stratocruiser simulator for BOAC. The rapid diffusion of jet aircraft highlighted the need to train pilots in a simulated environment to reproduce air turbulence and to practice recovery manoeuvres.

Simulators are used as synthetic devices in pilot training and checking programs to replicate aircraft behavior. Training programs and simulators are subjected to regulatory approval and certification requirements by public authorities. Credits of equivalency to live training can be earned only on certified simulators.

Live training has been almost totally substituted by simulator training because of (i) lower costs, (ii) reductions in training accidents, (iii) the possibility of practicing emergency recovery manoeuvres, and (iv) environmental issues. Typically, an experienced pilot will undergo a 120 hour program to complete the transition from one jet aircraft to another: (i) 50 hours of computer-based and classroom training, (ii) 40 hours with part task trainers and flight training device, (iii) 30 hours in a full-flight simulator, (iv) 15 hours of live training as a second in command.

Table 4 describes the four types of simulators. The *full-flight simulator* (FFS) is a full-size replica of a specific aircraft cockpit: mathematical models and computers simulate the behavior of the aircraft with original flight data, as well as the pilots' responses to varying

conditions. Realistic training for emergency manoeuvres or periodic checking requires faithful FFSs. High-level *flight training devices* (FTDs) basically have the same characteristics as FFS, although they are not necessarily equipped with motion or visual systems. FTDs are used for learning complex tasks such as flight management. *Generic training devices* are used for pilot instruction for small planes and for familiarizing pilots with the normal and emergency ground and air procedures. *Computer-based training devices* are used for the initial part of the training to gain familiarity with equipment and manoeuvres.

□ *A high level of interdependence in long-lasting, complex, customized transactions*

Simulators are not sold in high-volume but are customized products which entail intensive knowledge-based transactions. Air carriers contact reputed flight simulator developers and ask for bids. The bidding process lasts a few months and involves a substantial commitment of resources on the part of the bidders. During the bidding process, major development and engineering choices are made and the product is designed to meet the buyer's preferences as well as the regulator's expectations. The engineering process of design and integration lasts two years on average and involves intensive exchanges between the air carrier, the aircraft builder and the simulator manufacturer.

In the course of designing, building and maintaining simulators, many parties are thus involved. Simulators are customized to meet air navigation requirements and aircraft characteristics. Even within a given type of simulator, i.e. the 747-400, each airplane might be different, because air carriers tend to request different motors, avionics and cockpit displays. Training programs require distinct instructor stations, recovery scenarios, etc. Once delivered to the air carrier, the simulator is tested by the regulator for certification. In short, a complex set of interactions have to be performed for the development and installation of each simulator. The flight simulator industry is thus composed of sets of interacting parties which are highly interdependent, as described in Figure 1.

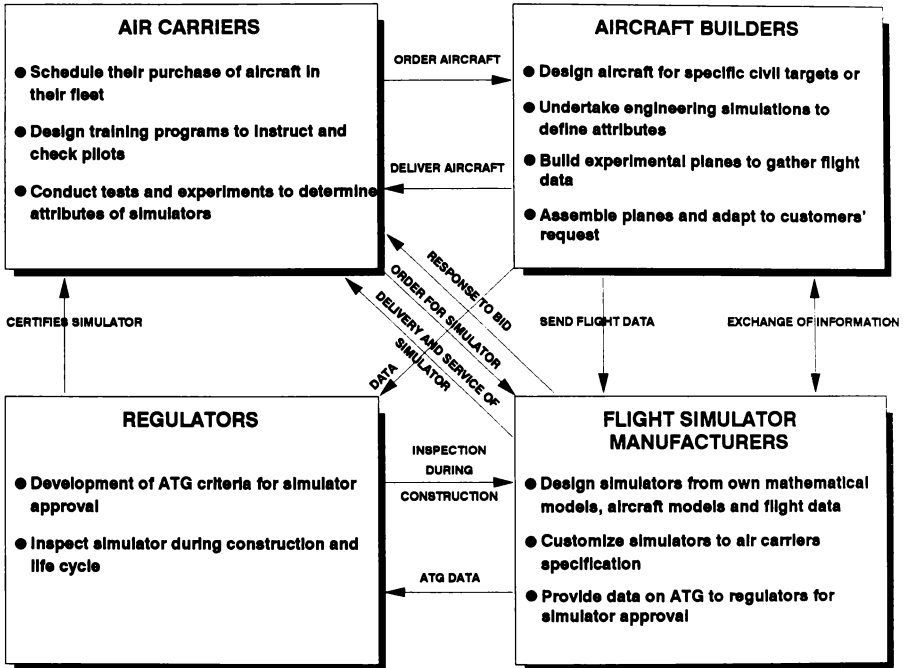


Figure 1. Parties forming the civil flight simulation meso-system

□ *A meso-system to negotiate the ex ante selection of major innovations*

Major innovations in flight simulation require the combined expertise of air carriers, systems developers, aircraft builders and regulators. These organizations form a meso-system. Each party wants innovations to fit within their existing systems, with a minimum of disruption. That and the need for zero-defect solutions make the pooling of expert knowledge and the practice of collective learning-by-using mandatory.

- Air carriers want reduced training costs, improved training effectiveness, and good relations with pilots. Major innovations raise three types of concerns: (i) Will innovative simulators increase training effectiveness and avoid negative training to such an extent that switching becomes worthwhile? (ii) Will the simulator be certified by the regulators? and (iii) Will a switch to the new technology render obsolete all prior investment in training programs and facilities?
- Simulation system developers want to develop innovations that meet their clients' expectations as well as regulatory approvals. No firm will invest to develop an innovative simulator without the assurance that it has a high probability of being certified and of finding a ready market.
- Aircraft builders are faced with lower uncertainties insofar as major innovations in flight simulation are concerned. The major impact on them is the cost of experimental flight data they will be asked to provide and the types of aerodynamic models they will need to divulge to simulation system developers.
- Regulators want to guide the field toward adopting innovations that really enhance safety or reduce costs without reducing safety. However, given the complexity of the technologies involved, regulators generally do not have the resources to direct the evolution of the field. They must rely on industry experts. Yet, at the same time, they "must keep their distance" from those experts, avoid stalling the flow of innovations and eliminate innovations that lead to negative training effects.

In sum, a meso-system has been crafted to negotiate and target major innovations *Ex ante*, rather than following the "hands-off" formula of letting firms embark on a variety of innovative experiments and then subjecting their results to the selective force of the market. The extent of participation of the main parties in the development of radical innovations is described in Table 5.

	RANK
• AIR CARRIERS	1
• AIRCRAFT BUILDERS	2
• SIMULATION SYSTEM DEVELOPERS	2
• AIR TRANSPORT INDUSTRY ASSOCIATIONS	3
• INTERNATIONAL BODIES (IATA, ICAO, ETC.)	3
• REGULATORS	4
• TRAINING ACADEMIES	4

Table 5. Level of participation

The key influencers of the innovation process in flight simulation belong to a relatively small set of highly interdependent organizations. As identified by our respondents (see Table 6) they come from such diverse groups as: air carriers (7), simulation system developers (5), aircraft manufacturers (3), regulators (2), and international or national industry associations (3).

b) The emergence of a global oligopoly

Over the years, three types of firms have evolved in this industry: (i) systems developers of FFS and FTDs; (ii) integrators of generic products; and (iii) specialists. Systems developers build prototypes customized to the specifications of air carriers. Integrators of generic training devices produce standard products for the general aviation industry or computer-based systems used in the early phase of pilot training. Specialized suppliers sell either components or services to members of the first two groups. The market dominance by the five largest system developers dates back to the late 60s, as Figure 2 indicates.

AIR CARRIERS	SIMULATOR SYSTEMS DEVELOPERS
American Airlines	CAE
Lufthansa	Rediffusion
United Airlines	Thompson
Northwest Airlines	Evans & Sutherland
British Airways	Link-Miles
KLM	
Delta Airlines	
AIRCRAFT MANUFACTURERS	REGULATORS
Boeing Aircraft	Federal Aviation
Airbus Industries	Administration (FAA) - U.S.A.
McDonnell-Douglas	Civil Aviation Administration (CAA) - U.K.
INTERNATIONAL BODIES AND ASSOCIATIONS	
Royal Aeronautical Society (RAeS)	
International Air Transport Association (IATA)	
Air Transport Association (ATA) - U.S.A.	

Table 6.

The industry exhibits a high degree of structural stability, especially within the system developers group. In fact, most of the competitors which entered the industry in the early 60s are still active, either as autonomous firms or as divisions of larger firms. Table 7 shows that the number of systems developers in 1992 was 14, down from a peak of 18 in 1985-86, and 10 in the early 60s. This early and continued process of concentration has resulted from a combination of (i) persistence of early entrants; (ii) a round of acquisitions of competitors by some system developers; and (iii) low level of late entries (only 3 major entries can be identified). The factors which explain the high levels of structural stability in this industry are:

i) *A non-stop learning race*

Simulation system developers are engaged in a never-ending learning contest to build expertise and credibility. The learning contest never ends because the industry's learning curve seems never to reach a plateau and because any learning-based advantage is highly appropriable and can easily become self-reinforcing (more on this below). System

developers thus compete primarily on the construction of dynamic capabilities and technical expertise in rapidly evolving fields such as training analysis, cognitive decisions making, mathematical modeling, software engineering and systems analysis. System developers also compete on the timely delivery of certified, high reliability simulators. Table 8 gives the rank ordering of the key elements of competitive strategies.

• SYSTEM DEVELOPERS OF FFSs AND FTDs	14
• INTEGRATORS OF GENERIC TRAINING DEVICES	12
• SPECIALIZED SUPPLIERS	
• COMPUTERS	13
• INSTRUMENTS	10
• MOTION SYSTEMS	5
• VISUAL SYSTEMS (HIGH AND LOW VALUE)	20
• TRAINING ANALYSES	12
• SOFTWARE HOUSES	10
• UPGRADING OF SIMULATORS	10
• OTHERS	9
	115
• DIVISION BY COUNTRY	
• U.S.A.	74
• U.K.	20
• FRANCE AND EUROPE	12
• CANADA	4
• JAPAN	2
• OTHERS	3
	115

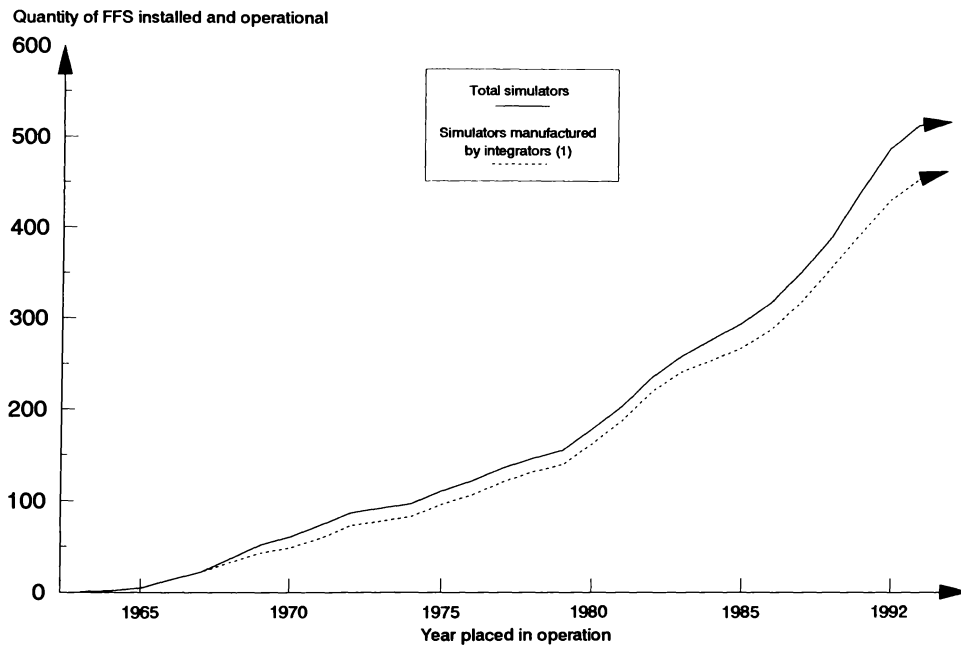
Source: Authors' compilations from ITEC, Aviation Week

Table 7. The flight simulator industry (1992)

	RANK
• LEARNING AND ACCUMULATION OF KNOWLEDGE	1
• BUILDING REPUTATION OF TECHNICAL EXPERTISE	2
• DELIVERY ON TIME	3
• CREDIBILITY GAINED BY PARTICIPATING IN WORKING GROUPS	4
• ECONOMIES OF SCALE AND COST REDUCTIONS	5
• PIONEERING INNOVATION	6

Table 8. Elements of the competitive strategies of systems developers

Figure 2. Dominance of sales by five system designers in civil sector



(1) Integrators are : 1) CAE
2) REDIFFUSION
3) THOMSON CSF

ii) Self-reinforcing processes

A self-reinforcing learning process is at work in the flight simulation industry: «The more simulators a firm sells, the faster it learns; the faster it learns, the more competitive it becomes and the more simulators it sells». This self-reinforcing learning process acts as a powerful selector: fast learners rise to market dominance, while slow learners are swept aside. Furthermore, the know-how gap between incumbents and potential newcomers becomes an increasingly formidable barrier to entry. Cumulative learning, therefore, has been a source of potent «early-mover advantages». Since the late 70s, potential entries have been deterred by the enormous amounts of resources to be committed before even the prospect of a sale contract.

Reputational effects are important because air carriers expect high levels of technical competence from system developers. Firms perceived to have the requisite accumulated expertise are invited to bid, while others are excluded from the process. Naturally, buyers prefer firms with accumulated knowledge because of their proven capacity to deliver quality products on time.

Potential entrants, even those with deep pockets, face quite a few years of profitless activities before breaking even. Aerospace firms, training centers, or FTD manufacturers, are generally perceived by air carriers as lacking the necessary technical credibility. The existence of high barriers to becoming a system designer is best illustrated by two considerations:

- ❑ Boeing Aircraft does not build flight simulators: it has expertise in aerodynamic simulations but not in training simulation. As a consequence, Boeing prefers not to integrate into flight simulation and instead relies on designers such as CAE, Rediffusion, Thomson-CSF, Link-Miles, etc.
- ❑ Specialist firms which assemble flight training devices may occasionally obtain contracts to build simulators which require high levels of fidelity. Often, such contracts end up in trouble because firms lack the systems engineering or software development skills to integrate parts into a systemic, operational whole.

c) Collaborating to compete

Cooperation and competition are traditionally viewed as mutually exclusive (Astley, 1985). Cooperation is conceived as creating conditions conducive to price fixing, exclusion of worthy newcomers and suppression of innovation, to the benefit of large incumbent firms.

Competition and collaboration strategies not only co-exist in the civil flight simulation field but are necessary complements of each other. "Collaborating to compete" summarizes succinctly the subtle approach which we have observed. Rivals in the flight simulation field collaborate to set rules and to develop anticipatory standards with users and regulators for major/radical innovations. However, the simulation system industry is fiercely competitive, once rules are set.

Cooperative strategies aim at developing *Ex ante* rules that (i) establish the anticipatory standards by which simulators will be evaluated, and (ii) that determine the functions of simulators in pilot training programs. While thus collaborating extensively, firms are involved in a competitive learning contest characterized by continuous investments to develop hard-to-imitate competencies and to remain flexible [40; 34]. Once prospective rules set the framework for innovation and establish anticipatory standards of certification, systems developers compete vigorously to develop the necessary innovations and to obtain sales orders.

Strategies of cooperation pursued by system developers focus both on (i) institutional rule-making, and (ii) long-term relations with forward-thinking clients as well as other parties in the flight simulation field. Table 9 describes the ranking we have obtained of the key elements of cooperative strategies. Networking and alliances are absent in the flight simulation field. In fact, system developers neither build stable suppliers networks nor engage in alliances. System developers do not want to establish stable links with suppliers. On the contrary, they prefer to be able to shift suppliers as technologies evolve and as air carriers request specific sub-systems.

	RANK
• LONG-TERM RELATIONS WITH INNOVATIVE CLIENTS	1
• RULE-MAKING AND ANTICIPATORY STANDARDS	2
• MEMBERSHIP IN WORKING GROUPS	3
• LONG-TERM RELATIONS WITH AIRCRAFT BUILDERS	4
• LONG-TERM RELATIONS WITH SUPPLIERS	5
• CREDIBILITY GAINED THROUGH PARTICIPATION	6

Table 9. Cooperative strategies in the flight simulation field

IV. Discussion: New Forms of Competition

More than 60 years after Allyn Young's presidential address to the American Economic Association, mainstream economists are finally catching up to his vision of an economy driven by increasing return mechanisms (Young, 1928). Young's world - and much of our world - is one of cumulative, self-reinforcing competitive processes, where small initial advantages can be amplified into long lasting market dominance [20; 23; 31].

An oligopolistic structure may emerge neither as the proportional result of inter-firm efficiency differentials (the so-called "Chicago view"), nor as a result of outright predatory practices (the "Harvard view"). Indeed, in Young's world, some of the old dichotomies (e.g., efficiency *versus* market power, predatory *versus* competitive practices, etc.) lose their relevance and even their meaning. In a world of increasing returns, market power *can* be extremely efficient. This is the point we want to emphasize next.

a) *In praise of tight oligopolies*

Until recent years, oligopolies have had few advocates among economists. True, long ago there was Schumpeter's (1947) valiant effort to give large, oligopolistic firms some of the credit they deserve for their role in the economic progress and dynamism of capitalist economies. More recently, Alfred Chandler - a historian respected and read by economists - has marshalled his considerable skills to give historical depth and resonance to the Schumpeterian message [8; 9]. By and large, however, only the current onslaught of "Chicago

economics" has awoken economists to the fact that perhaps oligopolies are not the malevolent, conspiracy-prone, inefficient beasts of old.

Unfortunately, the Chicago view of industrial economics has limitations of its own. Thus, while it correctly defends the right of surviving firms to enjoy the fruits of their harder work, smarter choices or greater luck, the Chicago interpretation of oligopoly formation has surrendered to a naive Darwinism, strictly meritocratic and deterministic, which sees the survival and dominance of the few as the optimal outcome of a "natural selection" process [2; 18; 13]. This may at times be so, but it need not always be so [19; 21].

Likewise, while "Chicagoans" have cogently argued that *actual* incidents of oligopolistic collusion should be the only concern of antitrust policy [6], in their extreme behaviorism they have not bothered to investigate what structural factors could make collusion likely, or unlikely. Fortunately, economists of other persuasions have conducted such a query. In recent years, their efforts have identified two types of markets where collusion is unlikely to take place, regardless of the level of industry concentration, namely, contestable markets and hyperselective markets.

Contestability theory is built upon two fundamental insights. The first one is that only sunk costs - that is, those costs which cannot be recovered upon exiting the market - are true barriers to entry. The second one is that in contestable markets - that is, in markets with negligible entry barriers - *potential* entry should suffice to encourage incumbents firms, regardless of their number, to refrain from collusion and to remain efficient. The corollary is clear: oligopolies, and even monopolies, will be efficient if they are contestable.

This is not the place to appraise the empirical relevance of contestability theory [37]. Let us just say that although the search for perfectly contestable markets is still on, the usefulness of the debate generated by this theory is beyond doubt: for the first time since Schumpeter's tour de force, a serious group of economists have dared suggest that an oligopoly - indeed, even a monopoly! - *can* be an efficient industry structure.

A more recent discovery, and one which is more pertinent to our study of the flight

simulation industry, is that of hyperselective markets, that is, markets subject to nonlinear increasing returns to market share and experience [20; 3; 29]. In such markets, due to the strength of increasing-return mechanisms (mainly scale effects, learning effects, reputation effects or network externalities), market dominance and market survival tend to go together. Firms have to dominate a markets in order to survive in them. Consequently, as long as the market keeps growing briskly, every firm has a strong incentive to vigorously compete for market *share* and not simply for market *space*. The risk of collusive arrangements becomes negligible, *even if entry barriers happen to be high* (as they are indeed in the flight simulation industry). In other words, increasing returns to market share and a steadily growing market will suffice to drive contenders to compete aggressively against each other, even if there are only two such contenders and even if the threat of new entrants is low.

Yet, even this acknowledgement of the potential for oligopolistic efficiency in industries where market dominance matters does not do full justice to some oligopolies. On the basis of what we have observed in the flight simulation industry, we feel compelled to not only defend them as *efficient* structures, but also to praise them as *effective* structures. This requires that we take a hard look at both the myth and the reality about inter-firm cooperation in oligopolistic industries.

b) In praise of cooperation for innovation

"It is important to understand how cooperation and competition are increasingly interwoven in modern industry. Our understanding of competition itself will be changed as we come to better understand the nature of cooperation" (Adler, 1989, p. 59).

Low-performance industries are all alike in that they all suffer from either a *coordination deficiency* or a *competition deficiency*, or both. Table 10 summarizes the simple relationships linking the levels of inter-firm coordination and competition to industry performance. Two points are worth stressing:

- first, coordination and competition are not, as is often implied, opposite ends along the same continuum. Rather, they each belong in a different, independent dimension;

- second, whereas coordination can only be deficient by default, competition can be deficient either by default or by excess, that is, by firms choosing to compete too aggressively (thus falling into a self-made trap of mutual destruction) or to collude and compete hardly at all.

LEVEL OF COORDINATION	LEVEL OF COMPETITION	
	HIGH	LOW
HIGH	CUT-THROAT COMPETITION	INSTITU- TIONAL DEADLOCK
MEDIUM	HIGH PERFORM- ANCE	INSTITU- TIONAL DEADLOCK
LOW	COLLUSION	INSTITU- TIONAL DEADLOCK

Table 10. Coordination, competition and industry performance

Current thinking about competition is far too dominated by the *pure rivalry view* of competitive dynamics. Implicit in this view is the notion that competing for clients in any markets is not unlike mining a lode of minerals close to the earth's surface. In such a situation, every rival would be best fending off for himself and any attempt to collaborate - e.g., in order to ensure a "reasonable" pricing regime - should be held as suspect, to say the least.

But not all industries fit this mold; certainly not at all times. In fact, most emerging industries and most so-called high-technology industries do not fit it at all. Competing in such industries is more like trying to settle a new territory than like trying to mine a very accessible mineral lode. In other words, in such industries, little can be accomplished without the various firms collaborating to build and enhance to requisite industry infrastructure, standards, coordinating procedures and reputation [32; 41].

"Industry takeoff" (that is, the process by which an industry moves into a stage of self-sustained growth) is a process not unlike that of an airplane's takeoff [41]. It generally is a complex, delicate and risky operation. As numerous examples have shown (e.g., the nuclear and videotex industries in the U.S., as

compared to those same industries in France), it takes more than simply an adequate technology for an industry to become "airborne". In addition, the institutional context has to be adequate too. This is where coordinating efforts - both among firms and between firms and governments - become crucial. It is coordination that converts a potentially negative-sum environment into a positive-sum environment. It is coordination that makes the difference between a botched up, aborted effort and a successful, fertile industry emergence. Clearly, for a promising, emerging industry, the preeminent risk to watch for is not simply the risk of incurring allocative inefficiencies, but rather the risk of institutional deadlock, typically due to a lack of vision and a lack of cooperation.

c) *The new institutional framework*

Our findings suggest a need to reconceptualize the dichotomy between cooperation and competition. Our central argument is that radical innovations in complex industries require first collaboration to determine *Ex ante* the appropriate standards and then competition to develop and market the innovations. Competition and collaboration are thus complements, not mutually exclusive opposites. In the absence of collaboration, high levels of symbiotic interdependencies can lock all parties into a deadlock type of situation because of coordination problems and high switching costs [22; 28]. Without a prior rule-making consensus, innovation would be limited to ingenious novel practices within a narrow range of options. No system developers would invest in R&D to design radically different designs without the certainty of acceptance by air carriers and regulators.

Safety concerns, however, also demand collaborative problem solving among public and private parties. Such concerns demand active guidance roles on the part of governments, as well as institutional mechanisms to ensure adequate collective action. Public pressures push governments to establish regulatory bodies to promote safety. Potentially contradictory roles are thus asked of governments. On the one hand, ill conceived regulations with a view to promote safety could lock-in an industry on old technologies. On the other, technical innovations are necessary to promote safety, but government is often a "blind giant" unaware of the technical complexity involved (David, 1990).

The paradox must however be resolved in practice if innovations are to materialize and become operational (Gillette, 1977). A hybrid system has emerged to suit the specific conditions of flight simulation where safety is crucial but where solutions cannot be reached without high levels of coordination among experts across specialized firms, high R&D investments, and the transformation of unmanageable uncertainty into manageable risk. The question is not whether governments should intervene, but rather how exactly should regulators perform their role (David, 1990; Farrell and Saloner, 1985). In this industry, regulators recognize that omniscience is impossible and accept to participate as an inquiring partner in sharing information, and in shaping anticipatory standards (Willen, 1983).

Institutional mechanisms were thus crafted in this industry to help develop innovations that promote safety while reducing training costs. This is not surprising. When technology is complex and when products are characterized by extensive learning in use, organizations to supplement pure market exchange must emerge (Lundvall, 1988; Reddy, 1988). Institutions are built to stimulate cooperation, to reduce uncertainty by information exchange and to ensure the trustworthiness of transacting parties (Lundvall, 1988; Lampel, 1992).

A process of collaborative decision-making has been observed in the flight simulation field. McCann (1983) has suggested that inter-organizational domains develop through sequential phases: (i) problem setting, (ii) direction setting, and (iii) institutional structuring.

i) *Problem setting* in flight simulation has typically been organized by legitimate convenors who bring interdependent stakeholders together. Problem setting with a view to collaborate has been the result of the actions of well-known individuals whose reputation and legitimacy made civilized inquiry and exchange possible. At the national level, FAA and the ATA have established joint study groups as issues emerge. Challengers of the established mindset have found a place to voice their innovations. Search conferences, working groups and formal experiments have been used to enlist participation. At the international level, the RAeS has taken the

role of informal convenor when issues transcend nations or when ICAO or IATA could not assemble rapidly the variety of parties to solve complex issues.

ii) *Direction setting and guidance* have been the result of informed discussions, search for preferred futures, and influence by the inner circle representing key executives from air carriers, regulatory agencies, systems developers and aircraft builders. Regulators used their certification powers to develop incentives in support of collective anticipatory standards. Publicly legitimated standards thus create the conditions of greater certainty and rewards necessary to justify innovation decisions, avoid under-investment in R&D (Tassey, 1982) and achieve a high level of coordination [7].

iii) *The structuring of institutional arrangements* has been a major achievement. The functions of institutional mechanisms are (i) to target innovative efforts, (ii) to ensure that major innovations are appropriate for all concerned parties before introduction, and (iii) to involve the regulator early in the process. The institutional mechanisms for collective decisions will be examined under five headings.

□ *Joint industry/government study groups to develop targeting rules*

Targeting means that working in joint industry/government study groups, representatives from the various parties negotiate and test worthwhile innovations. New rules are developed when shifts in the collective mindset make the introduction of radical changes feasible and acceptable. Regulators later officialize the guidelines by which new generations of simulators will be approved. Within this framework, simulation systems developers can invest in R&D to develop products meeting these targets.

□ *Debates and controversies*

Debates arise in the process of trying to resolve emerging technical controversies or to challenge the established mindset. Controversies arise because optimal solutions are neither unambiguous nor fully constraining. For instance, the initial questioning of the *zero-flight time mindset* was the result of studies by

NRC, IDA, Northwest Airlines and University of South Dakota. Challenges to the established mindset trigger participation from air carriers, flight simulator integrators and regulators. Eventually, industry-wide working groups are established to deal formally with the debates. A new collective mindset often emerges as issues are resolved.

□ *Working groups*

The functions of working groups are neither to determine technical choices in advance nor to identify future paths of innovation, but rather (i) to develop rules and regulations acceptable to parties, (ii) to debate technical issues, and (iii) to assess the probable impact on safety of the various innovations.

Professional bodies, such as the RAeS in London or AIAA in Washington, are the meeting places of the technical community. The RAeS in particular has emerged as the unofficial international coordinating body of the flight simulator sector. For instance, the working group on the international certification of simulators which included representative from FAA and other national regulators, air carriers, systems developers and aircraft builders was initiated and hosted by the RAeS. Eventually, its recommendations are officialized by national regulators.

□ *Guidance and active participation of the regulator*

A hybrid system has emerged in which industry self-regulation is shaped by proactive guidance by government agencies with rule-making authority. Regulators use their certification powers to support anticipatory standards that establish classification and performance attributes prior to the actual development of devices (David, 1990; Weiss, 1989). Conditions of greater certainty are thus created by the promulgation of anticipatory standards, making it possible for buyers to order artefacts and for simulation firms to develop them with the knowledge that certification will be forthcoming and that R&D investments will pay-off.

V. Conclusion

Past discussions of inter-firm coordination efforts were often marred by the implicit assumption that such a coordination can

only take the form of price collusion. More recent work, however, has been careful to avoid this error and to recognize that firms often coordinate things other than their price policies (e.g., their innovative activities) and that, unlike price collusion, such coordinating efforts are likely to increase public welfare [5].

Along these lines, our study of the flight simulation industry leads us to conclude:

- ❑ First, that in this industry at least, cooperation is in no way opposed to competition. On the contrary, flight simulation firms coordinate their efforts - in joint research activities, in identifying benchmarks and targets for future technical advance, in standard setting, etc. - precisely in order to better be able to compete, which they do vigorously. Cooperation is thus *pre-competitive* as well as *pro-competitive*.
- ❑ Second, in large part, the impressive performance of the flight simulation industry is *explained* by the fact that a very small, select and stable group of firms controls the world market. The precarious and subtle balance of pre-competitive cooperation and vigorous competition observed in this industry - and so necessary for its prosperity - would have been impossible to enact and to maintain if this industry had been populated by, say, 25 or 30 comparable rivals from all over the world. For one thing, members of a small, stable oligopoly have a higher incentive to refrain from "hit-and-run" competitive tactics (e.g., "dumping" in foreign markets, raiding the top engineering ranks of another firm, etc.). For another, firms in a tight oligopoly are less likely to trigger rounds of mutually destructive competition as a result of ignorance of each other's commitments, capabilities and expectations [32].
- ❑ Third, the various intra-firm and inter-firm resources, competences and coordinating procedures that have made such a high performance industry possible did not coalesce by chance or spontaneously. Specific individuals - in industry and in government - at specific times made it happen. Again, the contrast with the U.S. nuclear power industry is instructive in that it shows clearly that in high-technology craft industries, as in any other type of industry, neither success nor failure is foreordained by

the technical parameters of the situation. Technology merely opens doors and provides opportunity within a fairly permissive set of constraints.

- ❑ Fourth, while the above suggests that there is no single recipe for excellence and success whether for a whole industry or for a particular firm - this should not be taken to imply that successful industries can develop in any way whatsoever. In addition to the technical and commercial constraints proper to each industry, there is another inevitable constraint inherent in the very notion of "coordination equilibrium", namely, that of internal coherence. Whichever solution is found to the coordination problems faced by the members of an emerging industry, that solution must, by definition, be internally coherent. Otherwise, it will not coordinate, it will not hold, it will not be a lasting solution.
- ❑ Finally, while we know with certainty that the stable configuration and functioning of a successful industry must be internally coherent, we cannot know a priori which of the various configurations and behavioral patterns that are coherent and sustainable will end up materializing. Again, within a given set of broad constraints, there is much room left for historical contingency. Thus, for example, while the flight simulation industry could not possibly have emerged without its uncompromising commitment to high-quality products and to a cautious, consensus-driven, industry-wide innovation process, nothing in the parameters of the situation dictated that a single firm - CAE Industries - should come to hold a 60 % of the world market, nor that governments in the principal countries concerned would learn so rapidly to regulate this particular industry with a subtle, enlightened hand. This is why the emergence of a complex industry - like that of any complex, self-organizing entity - resist reductionist, deterministic interpretations. Not because there are no iron laws and fundamental principles in the domain of industrial economics, but rather because in spite of those laws and principles, so much is left to depend upon the occurrence of this or that contingent choice or event, especially in markets - like the flight simulation market - subject to increasing returns to market share.

References

1. ADLER, P.S. (1989). Technology Strategy: A Guide to the Literatures. Research in Technological Innovation, Management and Policy, 4; 25-151. Richard R. Rosenbloom ed., Greenwich, Conn.: JAI Press.
2. ALCHIAN, A.A. (1950). Uncertainty, Evolution and Economic Theory. *Journal of Political Economy*, 58(June); 211-22.
3. ARTHUR, W. B. (1988). *Self-Reinforcing Mechanisms in Economics*, in Philip W. Anderson and Kenneth J. Arrow, eds. The Economy as an Evolving Complex, Redwood City, California: Addison-Wesley; 9-31.
4. ARTHUR, W.B., ERMOLIEV, Y.M. and KANIOVSKI, Y.M. (1987). Path-Dependent Processes and the Emergence of Macro-Structure. *European Journal of Operational Research*, 30; 294-303.
5. BAUMOL, W.J., (1992). Horizontal Collusion and Innovation. *The Economic Journal*, 102 (January); 129-137.
6. BECKER, G.S. (1987). Antitrust's Only Proper Quarry: Collusion. *Business Week*, October.
7. CARLTON, D.W. and KLAMER, J.M. (1983). Coordination in Network Industries. *The University of Chicago Law Review*, 50; 446-465.
8. CHANDLER, A.D., (1977). "The Visible Hand". Cambridge, Massachusetts: The Belknap Press of Harvard University Press.
9. CHANDLER, A.D., (1990). "Scale and Scope: Dynamics and Industrial Capitalism". Cambridge, Massachusetts: The Belknap Press of Harvard University Press.
10. CLARK N. and CALESTOUS J. (1988). *Evolutionary Theories in Economic Thought*, in Dosi et al. (eds.), op.cit.; 197-218.
11. CORICELLI, F. and DOSI, G. (1988). *Coordination and Order in Economic Change and the Interpretative Power of Economic Theory*, in Dosi et al. (eds.), op.cit.; 124-147.
12. CUBBIN, J.S. (1988). "Market Structure and Performance - The Empirical Research". Alexis Jacquemin, ed., Harwood Academic Publishers, New York.
13. DEMSETZ, H. (1973). Industry Structure, Market Rivalry and Public Policy. *Journal of Law and Economics*, 16; 1-10.
14. DOSI, G. et al. (1988). Technical Change and Economic Theory. London: Pinter Publishers.
15. DOSI, G. and ORSENIGO, L. (1988). *Coordination and Transformation: An Overview of Structures, Behaviours and Change in Evolutionary Environments*, in Dosi et al., op.cit.; 13-17.
16. DRAZIN, R. and SANDELANDS, L. (1992). Autogenesis: A Perspective on the Process of Organizing. *Organization Science*, 3(2); 230-249.
17. FARRELL, J. (1986). A Note on Inertia in Market Share. *Economic Letters*, 21; 73-75.
18. FRIEDMAN, M. (1953). "The Methodology of Positive Economics." In *Essays in Positive Economics*. Chicago: University of Chicago Press.
19. HANSEN, R.G. and SAMUELSON, W.F. (1988). Evolution in Economic Games. *Journal of Economic Behavior and Organization*, 10; 315-338.
20. HENDERSON, B. D. (1979). "Henderson on Corporate Strategy". Cambridge, Massachusetts: Abt Books.
21. HODGSON, G.M. (1992). Thorstein Veblen and Post-Darwinian. *Cambridge Journal of Economics*, 16; 285-301.
22. HUGUES, T. (1983). "Networks of Power". Baltimore: Johns Hopkins University Press.
23. KALDOR, N. (1981). The Role of Increasing Returns, Technical Progress and Cumulative Causation in the Theory of International Trade and Economic Growth. *Economie Appliquée*, 4; 593-617.

24. LAMPEL, J. (1990). *Strategy in Thin Industries: Essays in the Social Organization of Industries*, Ph.D. Dissertation, Montréal, McGill University.
25. LESOURNE, J. (1985). Le marché et l'auto-organisation. *Economie Appliquée*, (3/4); 663-701.
26. LEWIN, R. (1993). "Complexity: Life at the Edge of Chaos". Macmillan, New York/Dent.
27. LOYE, D. and EISLER, R. (1987). Chaos and Transformation: Implications of Nonequilibrium Theory for Social Science and Society. *Behavioral Science*, 32; 53-65.
28. MILLER, R. and BLAIS, R. (1992). Configuration de modes d'innovation: les innovateurs prévisibles et les francs-tireurs. *Revue d'économie industrielle*, (61); 33-53.
29. OLLEROS, F.X. (1993). "Battles for Market Share in Hyperselective Markets", forthcoming in Jagdish N. Sheth, ed., Research in Marketing, 13; Greenwich, Conn.: JAI Press.
30. PANTZAR, M. (1992). Toward and Evolutionary View of Socio-Economic Systems. *World Futures*, 34; 83-103.
31. PORTER M.E. (1981). *Strategic Interaction: Some Lessons from Industry Histories for Theory and Antitrust Policy*, in Steven C. Salop, ed., Strategy, Predation and Antitrust Analysis. Washington, D.C.: Federal Trade Commission; 449-506.
32. PORTER M.E. (1980). "Competitive Strategy". New York: The Free Press.
33. RADZICKI, M.J. (1990). Institutional Dynamics, Deterministic Chaos, and Self-Organizing Systems. *Journal of Economic Issues*, XXIV(1); 57-102
34. RUMELT, R.P. (1984). *Towards a Strategic Theory of the Firm*, in Robert B. Lamb, ed., Competitive Strategic Management, Englewood Cliffs, N.J.: Prentice-Hall; 566-570.
35. SALAIS, R. and STORPER, M. (1992). The Four "Worlds" of Contemporary Industry. *Cambridge Journal of Economics*, 16; 169-193.
36. SHEPHERD, W.G. (1990). Mainstream Industrial Organization and «New» Schools. *Revue économique*, 41(3); 453-480.
37. SHEPHERD, W.G. (1984). "Contestability" vs. Competition. *The American Economic Review*, 74(4); 572-587.
38. SCHUMPETER, J.A. (1947). "Capitalism, Socialism and Democracy, New York: Harper.
39. SILVERGERG, G. (1988). *Modelling Economic Dynamics and Technical Change: Mathematical Approaches to Self-Organisation and Evolution*, in Dosi et al. (eds.), op.cit.; 531-559.
40. TEECE, D. ed. (1987). "The Competitive Challenge. Cambridge, Massachusetts: Ballinger Publishing Company.
41. VAN DE VEN, A.J. and GARUD, R. (1989). A Framework for Understanding the Emergence of New Industries. *Research on Technological Innovation, Management and Policy*, 4; 195-225.
42. WALDROP, M.M. (1992). "Complexity: The Emerging Science at the Edge of Order and Chaos". Simon and Schuster/Viking.

An Advanced Rotorcraft Flight Simulation Model: Parallel Implementation and Performance Analysis

S. Sarathy*

V.R. Murthy†

Abstract

Parallel processing techniques provide a powerful set of tools, which can permit the use of computationally complex models for real-time flight simulation. Elastic blade element models provide adequate fidelity and accuracy needed to simulate advanced rotorcraft configurations. Parallel implementations of blade element code will be needed to model such rotorcraft for real-time simulation applications. Techniques to develop flight simulation models for parallel (MIMD) architectures are presented in this paper. An advanced rotorcraft flight simulation model, using the blade element approach, has been developed for an MIMD parallel computer, based on a portable set of parallel primitives. Issues pertaining to this parallel development and implementation are also discussed. Parallel speedup results for this model are provided based on execution times on an Encore Multimax 14-processor machine.

Introduction

Meeting the demands of the next generation of advanced rotorcraft will require flight simulation models that are considerably more complex than those currently in use (Bailey's model and MBE model). Blade element (BE) models have been shown [1] to provide the fidelity and accuracy required by advanced rotor configurations, such as those in the AH-64, UH-60, MBB-105 and V-22 rotorcraft. BE models can easily incorporate a wide range of effects such as dynamic inflow, stall effects, compressibility, blade flexibility, bearingless root geometries and wake modeling. This *flexible fidelity* characteristic makes these models suitable for a wide range of rotorcraft flight simulation applications. However, the use of such models has been restricted to non-real time applications due to the heavy computational cost associated with these models. Fortunately, BE models possess a significant degree of inherent parallelism, allowing substantial speed up in the

performance of such flight simulation code, through the use of efficient parallel processing methods. The emergence of affordable second generation parallel computers, paves the way for large scale development of blade element flight simulation models.

In this paper, the parallel development of a rotorcraft flight simulation model is described. The blade element (BE) approach has been used to describe the rotor behavior. This work is similar to the parallel blade element model developed by Duval and Gillman [2, 3]. However, there are significant differences in both the model used and in the parallelization scheme adopted. The rotor model developed here, is based on an explicit second-order accurate nonlinear rigid blade equations with fully coupled flap-lag-pitch degrees of freedom. A six degree-of-freedom nonlinear model describing the response of the rotorcraft, is used to compute an explicit trim state and the response of the aircraft. The Duval-Gillman approach is based on a "physical component" model, wherein an explicit rotor blade or aircraft model does not exist. Instead a numerically computed set of coordinate transformations and component functions serve to define the state of the model. The parallel scheme adopted by Duval and Gillman, relies on task sharing at a higher level. This restricts the granularity of the parallelism realizable, thus confining this type of code to larger grained, shared memory MIMD architectures. In contrast, the approach used here, is based on exploiting parallelism in the model and in the numerical algorithms, at all levels. Therefore, such a parallel model can be adapted to a variety of parallel architectures.

This paper details the steps necessary to parallelize each phase of the flight simulation model. A basic set of parallel primitives, easily implemented on different MIMD architectures, is described. The use of these primitives in the development of the blade element model has been illustrated. Implementation issues for a shared memory MIMD version (Encore Multimax 14 processor) are also discussed. The parallel execution time metrics are presented for the Encore Multimax. This timing data is used to

*Research Fellow Georgia Institute Of Technology

†Associate Professor Syracuse University, Member AIAA

©American Institute of Aeronautics and Astronautics Inc. 1993. All rights reserved.

identify performance bottlenecks and subsequently refine the parallel model.

Flight Simulation Model

The primary functions of any flight simulation code are (i) aircraft trim, (ii) response and (iii) stability. The model incorporates these tasks using a blade element approach. The nonlinear trim involves the steady periodic solution, in the time domain, of the rigid blade equations of motion. The general aircraft equations of motion (6 dof) are linearized about the steady rectilinear flight trim state, to yield the small disturbance equations. This linear model is used to obtain the stability characteristics, handling qualities and the transient response of the rotorcraft for small changes in the state variables. These small disturbance equations are computed numerically, based on the general (nonlinear) aircraft equations of motion. The transient response is obtained through a local linearization scheme used to integrate the perturbation equations. These tasks have been characterized in the following table, based on the underlying mathematical problem, typical algorithms used to solve them and selected algorithms suitable for parallel processing.

Form	Method	Scheme
Trim		
Nonlinear fixed-point iteration	Newton-Raphson	Quasi-Newton
Response		
Nonlinear time integration	Explicit time marching	Linear model Integration
Stability and Handling Qualities		
Jacobian + Eigenvalue	Similarity, Norm reduction	Numerical Jacobian

The trim phase of the flight simulation model corresponds to finding a state vector which can minimize the resultant nonlinear vector function. This problem can be reduced to a fixed-point iteration and solved using Newton's iterative algorithms. For the current parallel implementation, a Quasi-Newton scheme is preferable based on its reduced computational requirements [4]. The response phase is represented by a time integration of the state equations. Several explicit time marching schemes have been used to perform this integration. However, for real time applications, a linearized model derived from the nonlinear state equations, are integrated instead. Based on this approach, a small disturbance model is derived for the aircraft. These equations require the computation of two jacobians, which are implemented numerically using a piecewise polynomial interpolation. The time

integration of this linear model is carried out using a first-order approximation of the exponential matrix. The stability information is obtained from the system and control derivatives (jacobians) and from an eigen-analysis of the linear model. The eigenvalue routines for real general matrices are based on either similarity transforms or norm reduction. A parallel version of a combined norm-reducing similarity transform scheme is available in [5].

Parallel Model Development

The primary issues in parallel code development are i) processor allocation, ii) synchronization, iii) mutual exclusion and iv) communication. For MIMD machines with shared memory, only the first three are important. Proper allocation ensures an equitable load distribution among all the processors and also determines the extent to which the parallelism in the code is utilized. Processors can be allocated either statically or dynamically. In the former, the allocation is done before the code is executed. In the latter, the actual allocation is based on the number of processors required and those available. Synchronization of processors is used to enforce data dependencies within the algorithm. Mutually exclusive access to memory segments is necessary to prevent potential data consistency, due to simultaneous write operations by two or more processors.

A parallel version of the flight simulation model can be obtained by introducing the appropriate set of parallel directives to enforce processor allocation, synchronization and mutual exclusion. However, each parallel computer has a different set of primitives to implement this functionality. This non-homogeneity in the parallel programming environment makes it difficult to develop general purpose parallel code, that can be ported to different machines with similar multi-processing architecture. One can mitigate this problem to a degree, by developing a set of basic parallel primitives which can be adapted to different parallel environments. This allows the application code to remain largely unchanged when it is ported. A basic set of these routines are presented in the table.

The `get/unget_process` pair provide process creation/termination control. They can be used to create new processes either statically or in a dynamic manner. These routines maintain a list of active processes and provides control of process allocation, at each nested levels, to the user. The `barrier` function is based on the usual `barrier` synchronization primitive, which forces all processors executing it, to idle-wait until a specified number of processors is reached. The `mutex` function is used to

Primitive	Function	System Calls
get_process	create	fork/create
	process/thread	/wakeup
unget_process	terminate	join/exit
	process/thread	/sleep
barrier	synchronize	barrier/spin_lock
		/semaphore/wait
mutex	mutually	spin_lock
	exclusive access	/comm.lock
pardo	parallel	IF-FOR
	DO loop	

isolate a critical memory segment, when single processor access is desired. The **pardo** structure is used when a DO- or FOR- loop can be unrolled and executed by several processors. This structure can be executed by one or more processors available, up to a user specified (static) limit.

Together, these primitives describe a parallel execution environment, which can be used on a variety of MIMD machines. The processes or *threads* are created or allocated from a pool of available processes. These parent and its child processes form a team, which together partition and execute the assigned parallel task(s). Typically, there is a greater degree of memory sharing within a team, than between teams. Each team member can form its own team, creating a hierarchy which closely mimics the task structure of the problem being solved. Since these teams could be formed dynamically, a mechanism to control such allocations is desirable. The user can define a (static) distribution of resources for each level of dynamic allocation. Each member of a team maintains a set of per-process (private) variables providing information on its identity (pid) and its relation to the team (teamid) and its team leader (parent). This information is used by the **barrier** primitive to synchronize members of a team and by the **pardo** structure to unroll DO-loops. These primitives can be used to develop the parallel model based on the algorithms chosen. Directed task tree graphs are extremely useful to identify data dependencies in the code. Based on these task trees for each algorithm used in the model, a preliminary parallel code can be generated.

Parallel Flight Simulation Model

The parallel flight simulation model developed here, uses a Quasi-Newton method for the trim phase^[4]. The response^[6] is based on the linearized model obtained from the small disturbance equations (SDE) about an equilibrium state (trim). The stability and

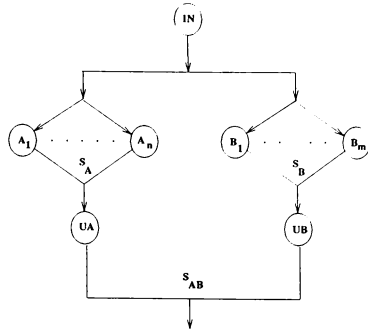


Figure 1: Directed Task Graph For SDE

handling qualities information is computed from the linear model state matrix and its eigenvalues. To illustrate the parallel code development, the algorithm, its task tree and corresponding parallel pseudo-code for the linearization (SDE) and response steps are given here.

Linearized Model: SDE

The numerical computation of the SDE is based on evaluating two jacobian matrices (**A** and **B**), corresponding to the state system and control matrices of the linearized system. Each jacobian computation, based on a p^{th} order approximation^[7], can be partitioned into $(p+1) \times N$ parallel sub-tasks, where N is the order of the jacobian. This can be seen in the task tree for this phase illustrated in Figure (1). In this flight simulation model, the sub-tasks refer to the evaluation of the net forces acting on the aircraft center of gravity, for a given state vector and control setting. The blade element approach adopted here, lends itself well to parallelizing each of these functional evaluations and hence can be sub-divided further.

For the task tree in Figure (1), $(p+1) \times N$ processes ($n = (p+1)N$ and $m = (p+1)M$) can be (statically) allocated to each of the sub-tasks (**A**₁ ... **A**_n and **B**₁ ... **B**_m). While the computations of net forces are independent (asynchronous) of each other, all the processes have to synchronize on completion of the sub-tasks at **S**_A and **S**_B, to assemble the jacobian matrix in step **U**_A and **U**_B. The processes from both jacobian computations, have to synchronize on exit from this procedure at **S**_{AB}. This task tree can be implemented using the primitives described earlier. Figure (2) outlines the basic steps involved, expressed in pseudo-code.

Algorithm For SDE

```

get_proc((p+1)(N+M))
Begin  $p^{th}$  order Jacobian
nproc_blk = (p+1) * (N+M)
Begin PARDO Loop ipar = 1, nproc_blk
  IF(pid = proc_list(ipar) AND
    ipar ≤ (p+1) * N) THEN
    Execute Task A
    j = (ipar - 1)/N + 1
    i = mod(ipar - 1, (p+1)) + 1
    Compute Trial Vector  $\tilde{x}_j$ 
    Evaluate Function  $F^i(\tilde{x}_j)$ 
     $S_A$ :Begin Critical: mutex_lock(lock1)
       $U_A$ :Update System Jacobian A
     $S_A$ :End Critical: mutex_unlock(lock1)
  ELSEIF(pid = proc_list(ipar)) THEN
    Execute Task B
    j = (ipar - 1 - (p+1) * N)/M + 1
    i = mod(ipar - 1 - (p+1) * N, (p+1)) + 1
    Compute Trial Vector  $\tilde{x}_j$ 
    Evaluate Function  $F^i(\tilde{x}_j)$ 
     $S_B$ :Begin Critical: mutex_lock(lock2)
       $U_B$ :Update Control Jacobian B
     $S_B$ :End Critical: mutex_unlock(lock2)
  ENDIF
End PARDO
barrier((p+1)(N+M))
End  $p^{th}$  order Jacobian

```

Figure 2: Implementation Of SDE Algorithm

In this code, the conditional branch (IF-THEN-ELSE), allocates the first $(p+1)N$ processes to task **A** and the remaining $(p+1)M$ to task **B**. Within each branch, the loop index $ipar$ is split into two indices i and j corresponding to a two-level nested loop. The outer index j has the range $[1, p+1]$, while the inner loop ranges from $[1, N]$ (or $[1, M]$). Each process is responsible for the computation of a net force vector **F** for different \tilde{x}_j , $(p+1)$ of which combine to yield a column of partial derivatives of the jacobian matrix. The synchronization within each branch is achieved by a `mutex_lock/mutex_unlock` pair around the critical jacobian update step. In this implementation, a single `mutex` is used for all the processes within the task. For large $(p+1)N$ or $(p+1)M$, this could lead to a high degree of memory contention and hence degrade the performance. In this case, instead of a single `mutex` lock, N (or M) separate locks can be used, limiting the number of processes per `mutex` to $(p+1)$. On completion of the matrix updates **UA** and **UB**, all processes synchronize at a **barrier** before exiting the procedure.

Response Integration

The integration of the linearized model is achieved through a first-order matrix approximation^[8, 9]. The task tree for this phase is given in Figure (3). Here, the primary tasks are the generation of the integration matrices (**E**, **P** and **Q**) and the dot-products **U**, **V** and **W**.

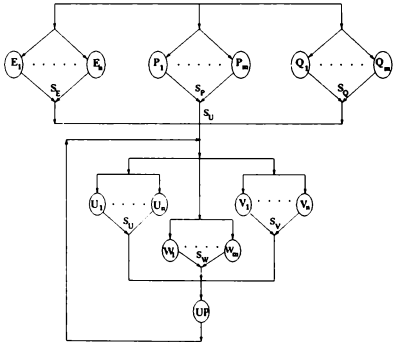


Figure 3: Directed Task Graph for Response

Since **E**, **P** and **Q** are explicitly defined in terms of the system (**A**) and control (**B**) jacobian matrices, they can be computed concurrently. The operations associated with these matrix evaluations are very fine-grained and require substantial synchronization. For a MIMD implementation, using a process to

compute each matrix element would involve a substantial synchronization penalty. Therefore, only N processes are allocated for each matrix computation, where N is the number of columns in the matrix. Thus, a total of $2N + M$ processes are allocated for this step. After all the columns of the matrix are computed, the processes are synchronized (S_E, S_P and S_Q) individually. The next task involves the response loop, wherein, the matrix products ($\bar{U}, \bar{V}, \bar{W}$) are computed iteratively. All processes are synchronized ($SEPQ$) before entering this loop. Within the loop, each matrix product is computed by N (or M) processes for the next time step and the state vector is updated in UP . The implementation of this task is shown in Figure (4). The $2N + M$ processes allocated for this task are partitioned by the PARD0 structure combined with the IF--THEN--ELSE statements. The barrier at the end of each sub-task ensures synchronization of all the processes within a team (S_E, S_P, S_Q). A second barrier ($SEPQ$) ensures that all three teams are synchronized prior to the response loop. A similar PARD0/IF--THEN--ELSE structure is used within the response loop to partition the processes for the sub-tasks U, V and W . The update of the state vector (UP) is performed by a single process while others wait at the next synchronization point (SUP). Since the update task is relatively fast, the performance degradation due to this sequential code fragment is acceptable. Another approach would be to use one or more mutex locks to update the state vector. A variant of this scheme incorporates the matrix evaluation task within the response loop. In this case the sub-tasks E, P and Q would be immediately followed by the matrix products W, U and V respectively. This would be followed by a global synchronization before the commencement of the next iteration. This implementation is depicted in the task tree given in Figure 5.

Results

The preliminary parallel performance results for the parallel model are extremely promising. Details of the parallel performance of the trim phase can be found elsewhere^[4]. Some timing results for the SDE and response phases are presented here. The timing runs were performed on a Encore Multimax machine. This computer has 14 processors and allows the creation of upto 32 asynchronous processes. The execution times for different tasks of the flight simulation model were measured for various static and dynamic allocations. The timing data presented here is in the form of relative speedup and processor effectiveness. The relative speedup measures the decrease in execution time of a n -processor run

Response Integration

```

Begin SDE Integration
nproc.blk = (p + 1) * (N + 2M)
Begin PARD0 Loop ipar = 1, nproc.blk
  IF (pid = proc_list(ipar) AND
      ipar ≤ N) THEN
    Compute E
    SE:barrier(lock1)
  ELSEIF (pid = proc_list(ipar) AND
      N < ipar ≤ N + M) THEN
    Compute P
    SP:barrier(lock2)
  ELSEIF (pid = proc_list(ipar) AND
      N + M < ipar ≤ N + 2M) THEN
    Compute Q
    SQ:barrier(lock3)
  ENDIF
SEPQ:barrier(lock0)
Begin Integration Loop
Begin PARD0 Loop ipar = 1, nproc.blk
  IF (pid = proc_list(ipar) AND
      ipar ≤ N) THEN
    Compute E $\bar{x}$ 
    ELSEIF (pid = proc_list(ipar) AND
      N < ipar ≤ N + M) THEN
    SW:barrier(lock1)
    Compute P $\bar{u}$ 
    SU:barrier(lock2)
    ELSEIF (pid = proc_list(ipar) AND
      N + M < ipar ≤ N + 2M) THEN
    Compute Q $\bar{u}$ 
    SV:barrier(lock3)
  ENDIF
  IF pid = parent THEN
    Update  $\bar{x} = E\bar{x} + P\bar{u} + Q\bar{u}$ 
  END IF
End Response Integration

```

Figure 4: Implementation Of Response Algorithm

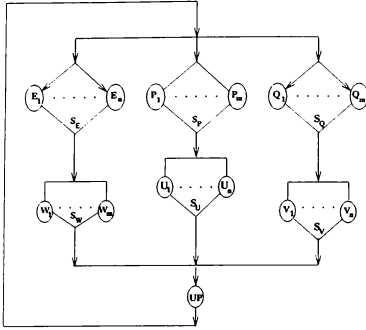


Figure 5: Directed Task Graph for Response Adaptive Algorithm

compared with that of a single processor run. The effectiveness serves as a descriptor of the overhead encountered by the parallel processors, due to synchronization and critical segments.

The SDE phase of the model achieves a significant level of speedup (Figure 6). Using fourteen processors, a maximum speedup of around 12 is achieved. The superior performance of this code is primarily due to the low synchronization cost associated with it. From the task graph for this algorithm (Figure 1), it can be seen that the tasks allocated to each process are independent of each other. Since each task involves a complete update of the aircraft's state, thus, the tasks are substantially larger than the overheads associated with the barrier synchronizations. Second, since all 14 processes are statically assigned to the top level tasks, there are no processors available for lower levels of dynamic process creation, therefore no overhead due to dynamic allocation.

The basic response scheme, without any adaptive integration, demonstrates a near-linear relation of speedup to number of processors. This speedup curves peaks at a speedup of 8 with 14 processes. The lower level of performance demonstrated by the response algorithm is due to two related factors. First, the size of the tasks allocated to each of the processes is much smaller than in the SDE case. Second, the tasks are primarily matrix operations and consequently require a finer level of synchronization. These two factors conspire to produce a relatively high overhead. This overhead rises with the increase in the number of processes, causing the peak and fall of the speedup curve. This

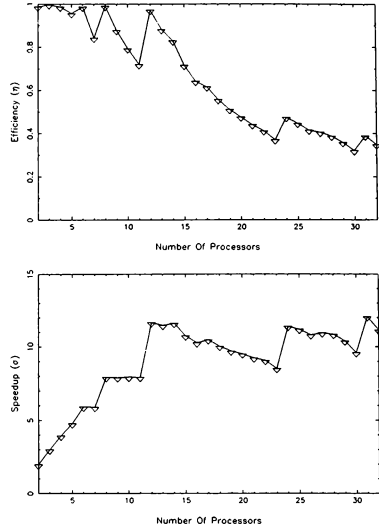


Figure 6: Parallel Speedup vs. N_{proc} : SDE

is clearly seen from the efficiency curve in Figure 7. Past $N_{proc} = 14$ the efficiency drops fast, signifying an increase in the synchronization overheads. The situation will improve when the task size increases. Therefore, the adaptive schemes proposed earlier, are likely to improve the performance of the response loop. This will be due to the reduction in the number of barrier calls as well as the increase in the concurrent task size.

Conclusion

This paper has detailed a method to develop flight simulation models for advanced rotorcraft using high fidelity blade element code. The use of parallel processing in such models has been shown to provide significant speedup over sequential code. Therefore, parallel blade element based rotorcraft models can provide the next logical step in the development of flight simulation.

Acknowledgements

The authors would like to thank Dr. Geoffrey Fox and Northeast Parallel Architecture Center for the use of their parallel computing facilities.

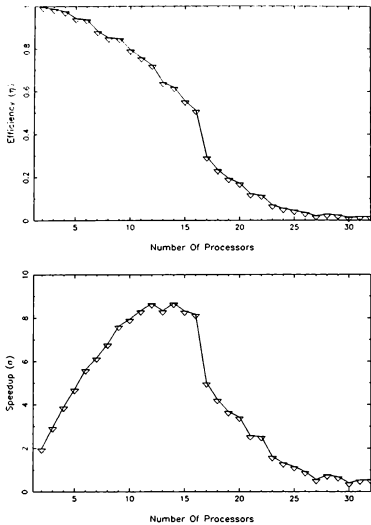


Figure 7: Parallel Speedup vs. N_{proc} Response

References

- [1] Lloyd Corliss, R.W. Du Val, H. Gillman, and L.C Huynh. *A Comparison Of Real-Time Blade-Element and Rotor-Map Helicopter Simulations Using Parallel Processing*. In 46th Annual Forum of the AHS, Washington D.C, May 1990.
- [2] R. Duval. *A Real Time Blade Element Helicopter Simulation for Handling Qualities Analysis*. In Fifteenth European Rotorcraft Forum, 1989.
- [3] H. Gillman. *A Parallel Architecture for a Real Time Blade Element Rotorcraft Simulation*. In Aerospace Simulation Conference. Society for Computer Simulation, 1988.
- [4] Sarathy, S. and Murthy, V. R. *Parallel Rotorcraft Flight Simulation*. In 31st Aerospace Sciences Meeting, Reno, jan 1993.
- [5] Shroff Gautam. M. *a parallel algorithm for the eigenvalues and eigenvectors of a general complex matrix*. Numerische Mathematik, 58:779-805, 1991.
- [6] S. Sarathy and V.R. Murthy. *An Advanced Parallel Rotorcraft Flight Simulation Model: Stability Characteristics and Handling Qualities*. In AIAA Aircraft Flight Mechanics and Flying Qualities Conference, Monterrey, CA, Aug 1993.
- [7] J. E. Dieudonne. *Description of a Computer Program and Numerical Technique for Developing Linear Perturbation Models from Nonlinear System Simulation*. Technical Report NACA TM 78710, NASA Langley Research Center, 1978.
- [8] J. A. Houck, F. L. Moore, J. J. Howlett, K. S. Pollock, and M. M. Browne. *Rotor Systems Research Aircraft Simulation Mathematical Model*. Technical Report TM 78629, NASA, November 1977.
- [9] J.A De Abrue-Garcia and T.T. Hartley. *Multistep Matrix integrators For Real Time Simulation*. Journal of Control And Dynamic Systems, 38:211-272, 1990.
- [10] Sarathy, S. *Advanced Rotorcraft Flight Simulation Model: Development, Validation and Parallel Implementation*. PhD thesis, Graduate School, Syracuse University, Syracuse, NY 13210, dec 1992.

METHODOLOGY FOR INTEGRATION OF DIGITAL CONTROL LOADERS IN AIRCRAFT SIMULATORS

Amnon Katz*, and Marc F. Schamlé†.

Department of Aerospace Engineering

The University of Alabama

P. O. Box 870280, Tuscaloosa, AL 35487-0280

ABSTRACT

A procedure is defined and demonstrated for programming a digital control loader in a flight simulator. The loader makes up for the difference in control force between simulator and aircraft. Aerodynamic forces, inertial effects and artificial feel systems in the aircraft are all included. Inertial effects and artificial feel systems in the simulator, if any, are accounted for.

I Introduction

In the early days of aviation (and to this day in light aircraft) pilot controls were coupled mechanically to aerodynamic control surfaces. Pilots relied on the control feel, dominated by aerodynamic forces, as a major cue. It is the function of the control loader to reproduce this cue in a flight simulator.

In the meantime aircraft have evolved. Hydraulically actuated controls have become the norm. Electronic controls are the trend of the future. These *irreversible* control systems do not feed aerodynamic forces back to the pilot. Artificial feel systems (usually springs) are used to provide the pilot with a semblance of the feel he expects. Increased reliance on instrument readings makes up for the deficiency.

Control loaders are fairly expensive. A high quality loader may cost more than a light airplane. This creates a paradoxical situation: a control loader can be financially justified only in those cases in which the most important cues that it can provide are suppressed. A very sophisticated piece of equipment ends simulating a mechanical spring, damper, and linkage. This is traditionally approximated by a near linear model [1].

In this paper we try to match the fidelity of the forces simulated to the sophistication of the device simulating them by addressing the exact non-linear geometry of a multi-stage mechanical link-

age. This is made practical by the advent of digital control loaders which allow the commanded force to be defined by software. Linearity is no longer an issue. Besides aerodynamic loads, we address inertial forces on any part of the control linkage, friction, springs in the aircraft and the simulator. The control loader is used to make up the difference between the force in the simulator and what it is in flight.

The procedure is as follows: Equations of motion of the control system are developed for the aircraft in flight and for the simulator in operation. The flight equations include:

1. Aerodynamic forces into the control linkage to the extent applicable.
2. Forces due to any artificial feel system.
3. Gravitational/inertial effects under the current local conditions of linear and angular accelerations due to maneuvering.
4. Pilot inputs.

The simulator equations include:

1. Control loader forces.
2. Forces due to any artificial feel system other than the control loader that may be retained in the simulator.
3. Gravitational/inertial effects under the current conditions of linear and angular acceleration due to motion base activity (or 1g).
4. Pilot inputs.

The effects of item 3 may differ in the simulator from what they are in the aircraft due to difference in mass properties of the simulated controls, absence of parts of the aircraft control linkage, and attenuation of the motion or, as in the case illustrated, absence of motion.

We postulate that the control motions and the pilot force inputs in the simulator are the same as in flight. Subtracting the corresponding aircraft and simulator equations, the unpredictable pilot inputs drop out. The resulting equations form the basis for programming the control loader.

The method is illustrated as applied to the longitudinal cyclic of the UH1 fixed base simulator at

* Professor, member AIAA.

† Graduate student.

the University of Alabama Flight Dynamics Lab. The control loader was added to the preexisting simulator as a part of a project on revalidation of flight simulators for civilian use. The purpose was to include analog equipment in the simulator, which was previously all digital. In this way revalidation issues of analog as well as digital systems could be addressed. The work on methodology of loader integration is an unplanned spinoff from the revalidation project.

As far as force cues, the UH1 uses a hydraulic system which, in normal operation, provides no feedback. Force feel is provided by springs and can be trimmed with the aid of a magnetic brake. The UA FDL simulator is constructed in a UH1-B fuselage and has the aircraft force feel and trim system built in. The aircraft systems were retained after integrating the loader. They generate the bulk of the force sensed by the pilot during normal operations. The control loader was used to generate the effects of flight acceleration on control linkage elements and the inertial and weight effects of such parts of the control linkage as are absent in the simulator. Whether these effects are significant in terms of generating realistic force cues and of meeting certification standards [2] remains to be determined. However, given the cost of the loader, the effort to program it correctly does not seem out of place.

Our method requires detailed information about geometrical and mass properties of the control linkage. We were not successful in securing this information from the manufacturer. We were forced to use our own rough estimates of these parameters.

The control loader also serves to simulate the increased control forces in case of hydraulic system shut-down or failure. Here again, we got no support from the manufacturer and were forced to resort to our own rough models.

II Formulation

Pilot controls and their linkage tend to be complicated mechanical systems. We use Lagrange's equations as the tool of analysis. This approach relies on a single scalar function - the *lagrangian* L - which is the kinetic energy expressed as a function of *generalized positions* q_i ($i = 1, \dots, n$) and their rates \dot{q}_i . The equations of motion are expressed in terms of $L(q, \dot{q})$ as

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = F_i, (i = 1, \dots, n). \quad (1)$$

F is the generalized external force, in terms of

which the element of work may be expressed as

$$dW = \sum_{i=1}^n F_i dq_i. \quad (2)$$

The effects of conservative forces may be included on the left side of equation (1) by including (minus) the potential in L . However, in the numerical computations that follow there is no advantage to doing so. We let L compute kinetic energy only and account for forces on the right hand side of (1). F represents the effects of friction, aerodynamic forces, inertial forces, spring loads in the artificial feel system, forces generated by the control loader, and forces applied by the pilot. The equations are stated twice: for flight and for the simulator:

$$\begin{aligned} \frac{d}{dt} \frac{\partial L^{flight}}{\partial \dot{q}_i} - \frac{\partial L^{flight}}{\partial q_i} = & F_{friction i}^{flight} + F_{aero i}^{flight} + F_{inertia i}^{flight} + F_{spring i}^{flight} + \\ & F_{pilot i}^{flight}, \quad (i = 1, \dots, n). \end{aligned} \quad (3)$$

$$\begin{aligned} \frac{d}{dt} \frac{\partial L^{sim}}{\partial \dot{q}_i} - \frac{\partial L^{sim}}{\partial q_i} = & F_{friction i}^{sim} + F_{loader i}^{sim} + F_{inertia i}^{sim} + F_{spring i}^{sim} + \\ & F_{pilot i}^{sim}, \quad (i = 1, \dots, n). \end{aligned} \quad (4)$$

We next subtract (3) from (4). The unknown pilot inputs cancel. We can now solve for the control loader input.

$$\begin{aligned} F_{loader i}^{sim} = & F_{aero i}^{flight} + F_{inertia i}^{flight} - F_{inertia i}^{sim} + \\ & F_{friction i}^{flight} - F_{friction i}^{sim} + F_{spring i}^{flight} - F_{spring i}^{sim} + \\ & \left(\frac{d}{dt} \frac{\partial}{\partial \dot{q}_i} - \frac{\partial}{\partial q_i} \right) (L^{flight} - L^{sim}). \end{aligned} \quad (5)$$

The index i goes over the degrees of freedom of the pilot control system. These would normally be longitudinal stick, lateral stick, rudder pedals, and throttle or collective ($n = 4$).

The application described here is restricted to the longitudinal cyclic - a single degree of freedom. We take our single independent variable to be the angular position θ of the cyclic stick in the longitudinal plane. The corresponding generalized force is the moment M measured at the stick hinge. Equation (5) becomes

$$\begin{aligned} M_{loader}^{sim} = & M_{aero}^{flight} + M_{inertia}^{flight} - M_{inertia}^{sim} + \\ & M_{friction}^{flight} - M_{friction}^{sim} + M_{spring}^{flight} - M_{spring}^{sim} + \\ & \left(\frac{d}{dt} \frac{\partial}{\partial \dot{\theta}} - \frac{\partial}{\partial \theta} \right) (L^{flight} - L^{sim}). \end{aligned} \quad (6)$$

III Modeling the Control Linkage

The mechanical control linkage of the UH1 consists of a series of belcranks (some of them attached to torque tubes) connected by pushrods. For the purpose of analysis the linkage is decomposed into "stages". Each stage consists of a control rod (1) connected to an arm (2) that rotates around a fixed hinge (Figure 1). Another arm (3) is rigidly attached to (2) and drives the next stage. In working out the geometry of each stage, it is necessary to know the data of arm (3) of the previous stage. This arm is shown in phantom in Figure 1 and labeled as item 0 of the stage. In this way item 3 of each stage becomes item 0 of the next stage. The first stage is driven by the cyclic stick.

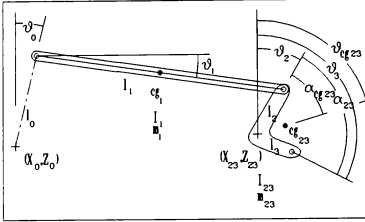


Figure 1: Generic Stage

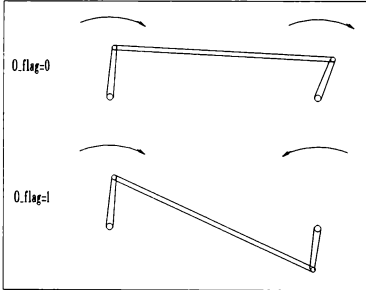


Figure 2: The Orientation Flag

The description of each stage includes mass properties: the mass and moment of inertia of item 1 and the CG location and combined moment of inertia of items 2 and 3. The mass properties of item 2 include those of item 3. The mass properties of item 0 are excluded, since they are accounted for in the previous stage.

The full description of a stage includes also an

orientation flag *O_flag* that specifies how items 0 and 2 are connected by item 1 (Figure 2). A value of zero means that items 0 and 2 rotate in the same direction. A value of 1 means they turn in opposite directions.

The control linkage of the longitudinal cyclic in the UH1 helicopter and in the University of Alabama Flight Dynamics Lab UH1 simulator are shown in Figure 3 (general view) and Figure 4 (stage assignment). The data is collected in Table 1.

We introduce the rotation θ_k^n and translation \bar{x}_k^n of member k of stage n . (The translation applies only to $k=1$.) These may be expressed in terms of θ_0^n and we introduce the following notations for these relationships:

$$\theta_k^n = \Theta_k^n(\theta_0^n), \quad (7)$$

$$\bar{x}_k^n = \bar{X}_k^n(\theta_0^n). \quad (8)$$

Θ_k^n and \bar{X}_k^n are worked out explicitly in Appendix A.

Table 1: Control Linkage Data

System	C	C	A	S	A
Stage No.	0	1	2	2	3
Note	1	2	2	2	3
$l_1(in)$	-	17.5	84.6	42.3	35.0
$l_2(in)$	26.3	2.25	3.00	5.5	3.0
$l_3(in)$	5.9	3.0	3.00	-	1.5
$x_{23}(in)$	0	-17.5	-102.1	-59.8	-102.1
$z_{23}(in)$	0	3.65	-2.35	6.15	-37.4
$\alpha_{23}(deg)$	168	180	-90	-	0
$m_1(lb)$	-	0.86	7.31	2.29	1.72
$m_{23}(lb)$	8.46	3.19	0.45	5.70	0.22
$\alpha_{cg23}(deg)$	-3	0	-45	0	0
$r_{cg23}(in)$	10.0	0.037	1.06	1.34	1.5
$I_1(lb \times in^2)$	-	21.9	4051	341	175.2
$I_{23}(lb \times in^2)$	2303	2.10	1.29	39.9	0.64

Legend: A - aircraft, S - simulator, C - common.

Notes:

1. Cyclic stick.
2. Controller arm.
3. Hydraulic servo control arm.

The kinetic energy of stage n may be expressed as

$$T_n(\theta_0^n, \dot{\theta}_0^n) = \frac{1}{2} [\dot{\theta}_0^n]^2 \left\{ I_1^n [\Theta_1^n(\theta_0^n)]^2 + I_{23}^n [\Theta_2^n(\theta_0^n)]^2 + m_1^n [\bar{X}_1^n(\theta_0^n)]^2 \right\}, \quad (9)$$

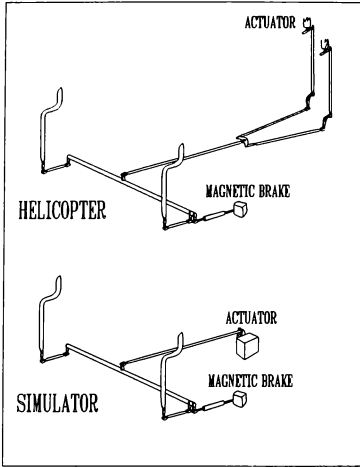


Figure 3:
Control Linkage of the Longitudinal
Cyclic in Helicopter and Simulator

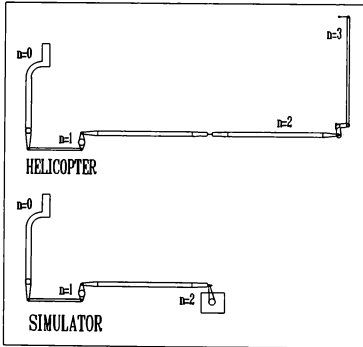


Figure 4: Stages of the Control Linkage

where I_k^n is the moment of inertia of the k component of stage n and m_k^n is its mass. The moment of inertia of belcrank arms is taken around their fixed axis of rotation, that of rods around their CG. I_2^n is the combined moment of inertia of arms 2 and 3.

Because of the rigid coupling between stages, we have

$$\theta_0^{n+1} = \theta_2^n. \quad (10)$$

The input into the whole linkage is a rotation $\theta_2^0 = \theta$ at the cyclic stick (Figure 5). We denote the relationship between the rotation of the k th component of the n th stage to the original input θ by B_k^n

$$\theta_k^n = B_k^n(\theta), \quad (11)$$

and the relationship between the translation of the k th component of the n th stage to the original input θ (used for $k = 1$ only) by \bar{D}_k^n

$$\bar{x}_k^n = \bar{D}_k^n(\theta). \quad (12)$$

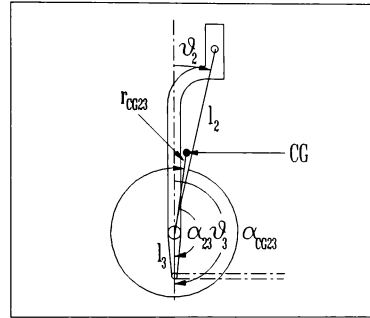


Figure 5: Stage 0 - the Cyclic Stick

The B_k^n may be obtained from the Θ_k^n by induction:

$$B_1^1(\theta) = \Theta_1^1(\theta), \quad (13)$$

$$B_k^n(\theta) = \Theta_k^n(B_3^{n-1}(\theta)). \quad (14)$$

A similar induction process produces $B_k^{n'}$, the derivative of B_k^n with respect to its argument

$$B_k^{1'}(\theta) = \Theta_k^{1'}(\theta), \quad (15)$$

$$B_k^{n'}(\theta) = \Theta_k^{n'}(B_3^{n-1}(\theta)) B_3^{n-1'}(\theta) \quad (16)$$

Similarly

$$B_k^{1''}(\theta) = \Theta_k^{1''}(\theta), \quad (17)$$

$$B_k^{n''}(\theta) = \Theta_k^{n''}(B_3^{n-1}(\theta)) [B_3^{n-1'}(\theta)]^2 + \Theta_k^{n'}(B_3^{n-1}(\theta)) B_3^{n-1'}(\theta) B_3^{n-1''}(\theta). \quad (18)$$

$\Theta_k^{n'}$ and $\Theta_k^{n''}$ are worked out in Appendix A.

A similar iteration scheme permits the construction of the \bar{D}_k^n from the \bar{X}_k^n :

$$\bar{D}_k^1(\theta) = \bar{X}_k^1(\theta), \quad (19)$$

$$\bar{D}_k^n(\theta) = \bar{X}_k^n(B_3^{n-1}(\theta)). \quad (20)$$

$$\bar{D}_k^{1'}(\theta) = \bar{X}_k^{1'}(\theta), \quad (21)$$

$$\bar{D}_k^{n'}(\theta) = \bar{X}_k^{n'}(B_3^{n-1}(\theta)) B_3^{n-1'}(\theta) \quad (22)$$

$$\bar{D}_k^{1''}(\theta) = \bar{X}_k^{1''}(\theta), \quad (23)$$

$$\begin{aligned} \bar{D}_k^{n''}(\theta) = & \bar{X}_k^{n''}(B_3^{n-1}(\theta)) [B_3^{n-1'}(\theta)]^2 \\ & + \bar{X}_k^{n'}(B_3^{n-1}(\theta)) B_3^{n-1'}(\theta) B_3^{n-1''}(\theta). \end{aligned} \quad (24)$$

$\bar{X}_k^{n'}$ and $\bar{X}_k^{n''}$ are worked out in Appendix A.

The kinetic energy of the control linkage is the sum of equation (9) over all stages. Substituting

$$\dot{\theta}_0^n = B_0^{n'}(\theta) \dot{\theta}, \quad (25)$$

expressing everything in terms of θ and $\dot{\theta}$, and summing, we find

$$T = \frac{1}{2} \dot{\theta}^2 G(\theta), \quad (26)$$

where

$$\begin{aligned} G(\theta) = & \sum_{n=0}^N \left\{ I_1^n [B_1^{n'}(\theta)]^2 \right. \\ & \left. + I_{23}^n [B_2^{n'}(\theta)]^2 + m_1^n [\bar{D}_1^{n'}(\theta)]^2 \right\}, \end{aligned} \quad (27)$$

When the Lagrange operator is applied to this we find

$$\left(\frac{d}{dt} \frac{\partial}{\partial \dot{\theta}} - \frac{\partial}{\partial \theta} \right) T = \bar{\theta} G(\theta) + \frac{1}{2} \dot{\theta}^2 G'(\theta). \quad (28)$$

The function G' which appears in (21) is given by

$$\begin{aligned} G'(\theta) = & 2 \sum_{n=0}^N \left\{ I_1^n B_1^{n'}(\theta) B_1^{n''}(\theta) \right. \\ & \left. + I_{23}^n B_2^{n'}(\theta) B_2^{n''}(\theta) + m_1^n \bar{D}_1^{n'}(\theta) \bar{D}_1^{n''}(\theta) \right\}, \end{aligned} \quad (29)$$

The difference of this expression for flight and simulation is to be substituted in equation (6).

The point of the foregoing manipulations is to show how all expression necessary in equation (6), including first and second derivatives of the angles may be obtained by purely algebraic operations based on the treatment of a single stage in

Appendix A. The actual computations have been carried out by a C program. The code uses a stage structure (object). At initialization, the program reads the data of all "stages" within the linkage from an input file and stores it in a linked list of stage objects. Then, for each simulation frame, the code goes over the linked list and accumulates $G(\theta)$ and $G'(\theta)$ at the same time as it inductively constructs the $B_k^n(\theta)$, $B_k^{n'}(\theta)$, $B_k^{n''}(\theta)$, and $vec D_k^n(\theta)$, $\bar{D}_k^{n'}(\theta)$, $vec \bar{D}_k^{n''}(\theta)$.

To improve throughput, the program does not actually compute values of angles, but only the sine and cosine of each angle. Appendix A shows how to connect the sine and cosine of each angle and the first and second derivatives of the angles in the terms of the input angle θ .

The loads due to inertia and friction are similarly accumulated. Appendix B contains the computation of the moments due to these effects for a single stage. The inputs are the geometry of the stage including location of hinge points in the longitudinal plane (x and z coordinates), the longitudinal load factors n_z and n_x , friction at hinges and rods, and for the last stage also air loads from the blade pitch horns, if any. The moment M_n is computed around the hinge of the arm, item n.0. Total moment around the cyclic stick (item 0.1) is accumulated by

$$M = \sum_{n=0}^N B_n'(\theta) M_n$$

The trim mechanism shown in Figures 3 and 4 forms a separate branch of the control linkage which is driven by stage 1. The treatment of this mechanism is similar in principle, although different in detail. It is omitted here for the sake of brevity. The significant contribution of this mechanism to stick force is the spring load. This, however, is identical in aircraft and simulator and drops out from equations (5) and (6).

IV RESULTS AND CONCLUSIONS

At the time of this writing the application described here has not yet been integrated in real time. Some typical values of commanded force computed by the formalism are presented in Table 2. For easy reference they are expressed in terms of force at the cyclic stick grip. These are the effects of the control loader as sensed by the pilot. The simulator is fixed base and operates at $n_x = 0$, $n_z = 1$. The load factors in the table refer to the flight condition being simulated. The table presents the total commanded force as well as the

increment due to maneuvering. It is seen that the forces are noticeable.

Table 2: Typical Forces Produced by the Control Loader

n_x	n_z	Force (lb) Total	Force (lb) Maneuver increment
0	1	0.58	0
0	3	0.73	0.15
0.5	1	3.09	2.52

The control loader is required to produce a force also when the cyclic stick is accelerated. 1g at the grip gives rise to 0.50lb contributed by the loader. This makes up for the inertia of control linkage parts absent in the simulator.

The magnitude of similar effects in other simulators will vary. Factors that tend to increase the forces required of the control loader are imbalance in the aircraft controls, imbalance in the simulator controls for simulators installed on motion bases, and omission of a large part of the aircraft linkage in the simulator.

The premise of flight simulation is that identical cues elicit identical responses. The aim of simulator engineering is to duplicate the flight environment to the level of accuracy that the pilot's senses can detect. The digital control loader makes it possible to pursue this goal in a more direct and precise manner than was previously practical.

Appendix A: GEOMETRY

A-1 Overview

This appendix treats the geometry of a single stage (Figure 1). It produces the angles θ_k ($k = 1, 2, 3$) and the displacement \bar{x}_1 and their derivatives in terms of the input angle θ_0 . The equations for Θ_k , Θ'_k , Θ''_k , and X_1 , X'_1 , X''_1 , which express θ_3 , $\frac{d\theta_3}{d\theta_0}$, $\frac{d^2\theta_3}{d\theta_0^2}$ and \bar{x}_1 , $\frac{d\bar{x}_1}{d\theta_0}$, $\frac{d^2\bar{x}_1}{d\theta_0^2}$ in terms of θ_0 are derived and made explicit. The stage index n is suppressed throughout the appendix.

The arms 0 and 2 in Figure 1 have one end hinged at (x_0, z_0) and (x_2, z_2) respectively. The other ends of these arms are at

$$(\xi_0, \zeta_0) = (x_0 - l_0 \sin \theta_0, z_0 - l_0 \cos \theta_0) \quad (A.1)$$

and

$$(\xi_2, \zeta_2) = (x_2 - l_2 \sin \theta_2, z_2 - l_2 \cos \theta_2). \quad (A.2)$$

These other ends are connected by member 1 and must therefore be a distance l_1 apart:

$$(\xi_2 - \xi_0)^2 + (\zeta_2 - \zeta_0)^2 = l_1^2. \quad (A.3)$$

This is the basic equation.

In the following eq. (A.3) is solved for θ_2 in terms of θ_0 . The derivatives of θ_2 with respect to θ_0 are readily obtained by taking the appropriate derivative of (A.3) with respect to θ_0 and solving for the desired derivative of θ_2 , freely using the previously derived expressions for θ_2 and its lower order derivatives. With θ_0 and θ_2 known, it is straightforward to obtain θ_3 , θ_1 , and \bar{x}_1 .

θ_3 will become θ_0 of the next stage. Since no angles appear in our equations except through their sine and cosine, we actually solve for the sine and cosine of each angle.

A-2 Solving for θ_2

When (A.1) and (A.2) are substituted in (A.3), the latter may be rewritten as

$$A \sin \theta_2 + B \cos \theta_2 = C, \quad (A.4)$$

where

$$A = 2(x_0 l_2 - x_2 l_2 - l_0 l_2 \sin \theta_0), \quad (A.5)$$

$$B = 2(z_0 l_2 - z_2 l_2 - l_0 l_2 \cos \theta_0), \quad (A.6)$$

$$C = l_1^2 - l_0^2 - l_2^2 - (x_0 - x_2)^2 - (z_0 - z_2)^2 - 2l_0[(x_2 - x_0) \sin \theta_0 + (z_2 - z_0) \cos \theta_0]. \quad (A.7)$$

Equation (A.5) in conjunction with

$$(\sin \theta_2)^2 + (\cos \theta_2)^2 = 1, \quad (A.8)$$

may be solved for $\sin \theta_2$ and $\cos \theta_2$. The results are

$$\cos \theta_2 = \frac{BC \pm A\sqrt{A^2 + B^2 - C^2}}{(A^2 + B^2)} \quad (A.9)$$

$$\sin \theta_2 = \frac{CA \mp B\sqrt{A^2 + B^2 - C^2}}{(A^2 + B^2)} \quad (A.10)$$

The last equations define two values of θ_2 . The choice of sign in (A.9) is tied to the value of the orientation flag *O-flag* and follows the rule:

$$\text{sign} = \begin{cases} \text{sign}(A \cos \theta_0 - B \sin \theta_0), & \text{O-flag} = 0, \\ \text{sign}(B \sin \theta_0 - A \cos \theta_0), & \text{O-flag} = 1. \end{cases} \quad (A.11)$$

A-3 Derivatives

Taking the derivative of equation (A.3) with respect to θ_0 and using (A.1) and (A.2) we find

$$(\xi_2 - \xi_0)(l_2 \cos \theta_2 \frac{d\theta_2}{d\theta_0} - l_0 \cos \theta_0) - (\zeta_2 - \zeta_0)(l_2 \sin \theta_2 \frac{d\theta_2}{d\theta_0} - l_0 \sin \theta_0) = 0. \quad (A.12)$$

This may be solved for $\frac{d\theta_2}{d\theta_0}$ as

$$\frac{d\theta_2}{d\theta_0} = \frac{l_0 (\xi_2 - \xi_0) \cos \theta_0 - (\zeta_2 - \zeta_0) \sin \theta_0}{l_2 (\xi_2 - \xi_0) \cos \theta_2 - (\zeta_2 - \zeta_0) \sin \theta_2}. \quad (A.13)$$

Expressions for (ξ_0, ζ_0) and (ξ_2, ζ_2) are provided in (A.1) and (A.2). Expressions for $\sin \theta_2$ and $\cos \theta_2$ in terms of θ_0 are provided in the last section. When all those are substituted in (A.17), $\frac{d\theta_2}{d\theta_0}$ is expressed in terms of θ_0 .

Taking the derivative of (A.12) with respect to θ_0 (this is the second derivative of (A.3)) we find

$$\begin{aligned} & (\xi_2 - \xi_0) \left(l_2 \cos \theta_2 \frac{d^2 \theta_2}{d\theta_0^2} - \right. \\ & \quad \left. l_2 \sin \theta_2 \left(\frac{d\theta_2}{d\theta_0} \right)^2 - l_0 \sin \theta_0 \right) - \\ & (\xi_2 - \zeta_0) \left(l_2 \sin \theta_2 \frac{d^2 \theta_2}{d\theta_0^2} + \right. \\ & \quad \left. l_2 \cos \theta_2 \left(\frac{d\theta_2}{d\theta_0} \right)^2 + l_0 \cos \theta_0 \right) + \\ & (l_2 \cos \theta_2 \frac{d\theta_2}{d\theta_0} - l_0 \cos \theta_0)^2 + \\ & (l_2 \sin \theta_2 \frac{d\theta_2}{d\theta_0} - l_0 \sin \theta_0)^2 = 0 \end{aligned} \quad (A.14)$$

This may be solved for $\frac{d^2 \theta_2}{d\theta_0^2}$ as

$$\frac{d^2 \theta_2}{d\theta_0^2} = \frac{P}{Q}, \quad (A.15)$$

where

$$\begin{aligned} P = & (l_2 \cos \theta_2 \frac{d\theta_2}{d\theta_0} - l_0 \cos \theta_0)^2 + (l_2 \sin \theta_2 \frac{d\theta_2}{d\theta_0} - l_0 \sin \theta_0)^2 + \\ & (\xi_2 - \xi_0) (l_2 \sin \theta_2 \left(\frac{d\theta_2}{d\theta_0} \right)^2 - l_0 \sin \theta_0) - \\ & (\xi_2 - \zeta_0) (l_2 \cos \theta_2 \left(\frac{d\theta_2}{d\theta_0} \right)^2 + l_0 \cos \theta_0), \quad (A.16) \\ Q = & l_2 [(\xi_2 - \xi_0) \cos \theta_2 - (\zeta_2 - \zeta_0) \sin \theta_2]. \quad (A.17) \end{aligned}$$

Expressions for all the quantities on the right hand side of (A.16) and (A.17) in terms of θ_0 have been provided.

A-4 θ_3 and derivatives

Member 3 of each stage is a rigid extension of member 2.

$$\theta_3 = \theta_2 + \alpha_{23}, \quad (A.18)$$

(see Figure 1). The sine and cosine of θ_3 are obtained by the trigonometric identities for the function of the sum of two angles:

$$\sin \theta_3 = \sin \theta_2 \cos \alpha_{23} + \cos \theta_2 \sin \alpha_{23}, \quad (A.19)$$

$$\cos \theta_3 = \cos \theta_2 \cos \alpha_{23} - \sin \theta_2 \sin \alpha_{23}, \quad (A.20)$$

Also, from (A.18), we have

$$\Theta'_3(\theta_0) \equiv \frac{d\theta_3}{d\theta_0} = \frac{d\theta_2}{d\theta_0}. \quad (A.21)$$

$$\Theta''_3(\theta_0) \equiv \frac{d^2 \theta_3}{d\theta_0^2} = \frac{d^2 \theta_2}{d\theta_0^2} = \frac{P}{Q}. \quad (A.22)$$

In this way Θ_3 , Θ'_3 , and Θ''_3 have all been reduced to algebraic expressions in $\sin \theta_0$ and $\cos \theta_0$.

A-5 Member 1

At this point $\xi_0, \zeta_0, \xi_2, \zeta_2$ are all known. The angle of member 1 is readily determined as

$$\sin \theta_1 = \frac{\zeta_0 - \zeta_2}{l_1}, \quad (A.23)$$

$$\cos \theta_1 = \frac{\xi_2 - \xi_0}{l_1}. \quad (A.24)$$

The CG location of this member is assumed at the center, which yields

$$\bar{x}_1 = \frac{1}{2} \begin{pmatrix} \xi_0 + \xi_2 \\ \zeta_0 + \zeta_2 \end{pmatrix}. \quad (A.25)$$

The derivatives of these quantities follow directly:

$$\begin{aligned} \frac{d\theta_1}{d\theta_0} = & \pm l_1^{-1} \left\{ (l_2 \sin \theta_2 \frac{d\theta_2}{d\theta_0} - l_0 \sin \theta_0)^2 \right. \\ & \left. + (l_2 \cos \theta_2 \frac{d\theta_2}{d\theta_0} - l_0 \cos \theta_0)^2 \right\}^{\frac{1}{2}}, \quad (A.26) \end{aligned}$$

with the sign being the same as the sign of

$$\cos \theta_1 (l_2 \sin \theta_2 \frac{d\theta_2}{d\theta_0} - l_0 \sin \theta_0). \quad (A.27)$$

$$\begin{aligned} \frac{d^2 \theta_1}{d\theta_0^2} = & \pm l_1^{-1} \times \\ & \left\{ \left[l_2 \sin \theta_2 \frac{d^2 \theta_2}{d\theta_0^2} + l_2 \cos \theta_2 \left(\frac{d\theta_2}{d\theta_0} \right)^2 - l_0 \cos \theta_0 \right]^2 + \right. \\ & \left. \left[l_2 \cos \theta_2 \frac{d^2 \theta_2}{d\theta_0^2} - l_2 \sin \theta_2 \left(\frac{d\theta_2}{d\theta_0} \right)^2 + l_0 \sin \theta_0 \right]^2 \right\}^{\frac{1}{2}} \quad (A.28) \end{aligned}$$

with the sign being the same as that of

$$\sin\theta_1 \left[l_2 \sin\theta_2 \frac{d^2\theta_2}{d\theta_0^2} + l_2 \cos\theta_2 \left(\frac{d\theta_2}{d\theta_0} \right)^2 - l_0 \cos\theta_0 + l_1 \left(\frac{d\theta_1}{d\theta_0} \right)^2 \right] \quad (A.29)$$

$$\frac{d\bar{x}_1}{d\theta_0} = \frac{1}{2} \left(l_0 \cos\theta_0 + l_2 \cos\theta_2 \frac{d\theta_2}{d\theta_0} - l_0 \sin\theta_0 + l_2 \sin\theta_2 \frac{d\theta_2}{d\theta_0} \right) \quad (A.30)$$

$$\frac{d^2\bar{x}_1}{d\theta_0^2} = \frac{1}{2} \left(\begin{array}{c} -l_0 \sin\theta_0 - l_2 \sin\theta_2 \frac{d\theta_2}{d\theta_0} \\ + l_2 \cos\theta_2 \frac{d^2\theta_2}{d\theta_0^2} \\ -l_0 \cos\theta_0 + l_2 \cos\theta_2 \frac{d\theta_2}{d\theta_0} \\ + l_2 \sin\theta_2 \frac{d^2\theta_2}{d\theta_0^2} \end{array} \right) \quad (A.31)$$

Appendix B: LOADS ON A STAGE

B-1 Inertial load

The inertial load contributed by a stage under load factors n_x (longitudinal) and n_z (vertical), expressed as a moment at the cyclic stick is

$$M_0 = m_1 g [n_x D'_{x1}(\theta_0) + n_z D'_{z1}(\theta_0)] + m_{23} g r_{cg23} B'_{23} [n_x \sin\theta_{cg23} - n_z \cos\theta_{cg23}] \quad (B.1)$$

In the above equation X_1, Z_1 are the x and z components of \bar{X}_1 ; θ_{cg23} is given by

$$\theta_{cg23} = \theta_2 + \alpha_{cg23}, \quad (B.2)$$

however, true to the policy of not computing trigonometric functions in real time, we express its sine and cosine as

$$\sin\theta_{cg23} = \sin\theta_2 \cos\alpha_{cg23} + \cos\theta_2 \sin\alpha_{cg23} \quad (B.3)$$

$$\cos\theta_{cg23} = \cos\theta_2 \cos\alpha_{cg23} - \sin\theta_2 \sin\alpha_{cg23} \quad (B.4)$$

B-2 Friction

Friction is assumed at each rotation hinge. The frictional moment at the 2 hinge is transferred to the cyclic stick by multiplying by $B'_2(\theta_0)$. When the linkage is in motion the friction force is applied in the direction opposing the motion. When the linkage is stationary, no motion command is sent to the control loader actuator unless the commanded force exceeds the friction.

B-3 Air loads

In irreversible control systems, the 23 element of the last stage is a control surface or a helicopter blade with its pitch arm. The aerodynamic force is then introduced as a moment on that stage and transformed to a moment at the stick by multiplying by $B_2^{N'}(\theta)$, where N is the number of the last stage.

ACKNOWLEDGEMENTS

This work was supported by grant 92G0012 from the Federal Aviation Administration. The FAA Grant also supported 50% of the cost of acquiring the McFadden model 192B control loader. The balance was supported by the James R. Cudworth Endowment at the University of Alabama. The authors wish to thank Dr. Jack Dohme and Major William Barker of the U.S. Army Research Institute Army for access to data on the UH1. The help of Dr. K. Krishnakumar in securing data is also appreciated.

REFERENCES

1. J. M. Rolfe and K. J. Staples, *Flight Simulation*, Cambridge University Press, Cambridge 1986, pp77-81.
2. "Airplane Simulator Qualification," U.S. Department of Transportation, Federal Aviation Administration, Advisory Circular 120-40B, July 29, 91.

DYNAMIC SIMULATION FIDELITY IMPROVEMENT USING TRANSFER FUNCTION STATE EXTRAPOLATION

William J. Bezdek*
Principal Technical Specialist

Larry A. Moody**
Technical Specialist

McDonnell Douglas Aerospace

Abstract

To achieve maximum dynamic fidelity in a real time manned flight simulator, it is imperative that any hardware or computational delays be reduced as much as possible. Unfortunately, some visual systems continue to have a significant processing time delay. To compensate for these delays, general form extrapolation schemes have been employed by the simulation industry. This paper presents a transfer function extrapolation method which takes advantage of the second order nature of most aircraft models to extrapolate aircraft state with greater accuracy and at higher input signal frequencies. To be of practical benefit to the simulation community, issues of commonality and implementation have also been addressed. To allow this approach to be independent of aircraft type, a method to calculate the transfer function damping and natural frequency from the past values of the original aircraft model is presented.

Nomenclature

δ_{ϕ}	roll rate command, degrees/sec.
ϕ	roll position, degrees
ϕ_m	magnitude of peak output signal
$\dot{\phi}$	roll rate, degrees/sec.
$\ddot{\phi}$	roll acceleration, degrees/sec. ²
ϕ_{ex}	extrapolated roll position, degrees
$G(s)$	second order transfer function
$H(s)$	feedback transfer function
M	input signal magnitude
ω	undamped natural frequency, rad/sec.
ω_f	input signal frequency, rad/sec.
τ	time constant, nondimensional
ζ	damping, nondimensional
ψ_f	phase angle, radians

Background

USAF MIL F 8785C requires that transport delay for flight hardware systems be less than 100 milliseconds.

onds, so this number has also become the industry objective for flight simulators. With the significant increase in computer processing speed and multi-processor capability, it has become possible to execute the real time loop of current generation high fidelity aircraft simulations at rates of up to 10 times faster than those of the previous generation of simulations, much faster than the MIL F 8785C requirement. This increase in update rate is particularly desirable in simulations which evaluate the aircraft dynamic response to high frequency inputs, such as in the development and testing of flight control systems. Modeling the "analog" aircraft dynamics at these faster rates eliminates many of the errors due to the digital nature of the model.

Unfortunately, the time required to draw the visual scene has not been proportionally reduced. In fact, the visual system delay is often higher than the time required to process several frames of aircraft data. Eliminating visual delays is especially important to pilot-in-the-loop flying qualities evaluations. In these evaluations, the pilot provides input commands to the aircraft based on the feedback that he receives from the visual scene. If the visual scene is delayed, the aircraft handling characteristics will be corrupted from the pilot's perspective, especially in fixed-based simulations.

Compounding this problem is the fact that the resolution of high fidelity simulations is increasing to the extent that high frequency lag effects can be modeled with significant accuracy. These effects include structural flexibility and control system lag filters. Most of the previous generation simulations had not modeled the the higher frequency delays since the transport delays from the simulation itself were approximately the same magnitude as the expected aircraft lags. However, it is desirable to model these extra delays whenever it can be done accurately, since it allows control system designers to use the simulation to test their control system over greater bandwidth frequencies.

To fully realize the benefit of the faster host update rates and increased model fidelity, an effort was initiated to take advantage of the excess processing

* AIAA Senior Member

**AIAA Member

capability of the host computer (compared to the time to update one frame of the visual scene) to extrapolate the aircraft state. The goal was to project aircraft state ahead in time by the amount of time taken to process the visual scene. Experience with existing general form extrapolators that resided in the visual system preprocessor had not been positive due to the lack of robustness of the methods to variances in the arrival time of new data across a network.

Approach

The initial approach proposed was to simply create a "second" aircraft model that would execute in parallel with the primary aircraft model. The "second" aircraft would be integrated with a slightly larger time step than the primary aircraft model, so that the aircraft state would be valid at the end of the visual system update rather than at the end of the aircraft update. Different data streams would be maintained for each aircraft model, to insure that integration of first and second derivative variables (such as velocity and acceleration) would be consistent with the appropriate time step and appropriate past values. However, past values needed by the extrapolation model came from the primary model, as they were available.

To avoid the complications of using a duplicate of the specific primary model for extrapolation, a second order transfer function model was implemented for each axis. This transfer function approach makes the assumption that the dynamics of each axis can be modeled by a nonhomogeneous second order differential equation of the form:

$$\phi'' + 2\zeta\omega\phi' + \omega^2\phi = \alpha(t),$$

where ζ and ω are the damping and undamped natural frequency, and α is the forcing function, which in turn is a function of time. The assumption that each axis can be modeled by a second order system is an excellent assumption, and is widely accepted in industry. However, it is clear that this approach by itself cannot adequately resolve highly cross coupled dynamics. Fortunately, the cross coupled effects are part of the last pass aircraft state that is used to initialize the transfer function model each pass, which demonstrates the value of calculating the transfer function coefficients in real time.

From the differential equation of the system, it is possible to derive a transfer function that relates the

system output to its input by taking the LaPlace transform of the system characteristic equation. The result is $G(s)$ shown in Fig. 1. Note also the LaPlace transform of the lead filter, which gives the system its extrapolation characteristics.

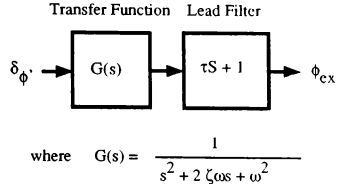
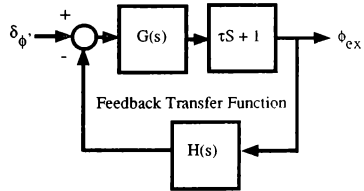


Figure 1. Extrapolation of Roll

An algorithm was then derived to calculate the coefficients of the transfer function from the past values of the input and output of the primary aircraft model. At this point the extrapolation model is completely independent of the primary model and self initializing. It can therefore be used without modification for any second order aircraft or missile.



where $H(s) = gs$, g = variable gain

Figure 2. Extrapolator With Feedback

To obtain additional accuracy, the transfer function is allowed to iterate several passes after being updated with each set of fresh data from the primary aircraft model. A negative feedback loop, with a variable feedback gain, modifies the system input for each subsequent iteration, allowing the transfer function to converge more accurately on the expected response. The time step used for integration of the transfer function is then slightly greater than

the aircraft model frame time divided by the number of transfer function iterations as shown in Fig. 2.

Rather than using the roll rate command that is associated with the last aircraft state, it is now possible to use the most current value of roll rate command, directly from the crew station. This is possible since the last aircraft state is simply used to create the transfer function, and is independent of new inputs. The transfer function can be thought of as a complete response model. The transfer function integration into the simulation is shown in Fig. 3.

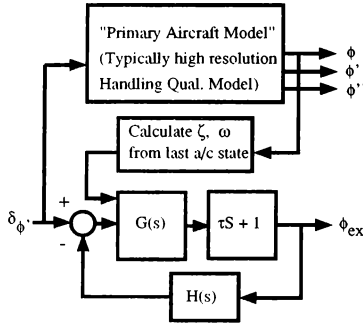


Figure 3. General Arrangement

At this point, it becomes clear that there are several practical ways to implement this extrapolator. One of the more interesting approaches is to place the transfer function logic in the visual system processor and feed fresh crew station inputs to the visual system directly (while simultaneously feeding the same crew station inputs to the primary aircraft model in the host computer). This approach reduces the cycle time to simply that required to asynchronously read the crew station hardware, update the transfer function, and draw the visual scene. With extrapolation effects, the effective transport delay could be close to zero, for mid-range input frequencies. If this approach is implemented, it is recommend that the transfer function feedback loop be eliminated, or at least solved with fewer iterations, to help reduce the time required to update the transfer function.

To allow straightforward calculation of the transfer function coefficients from the last aircraft state,

the assumption was made that the primary aircraft model had converged in response to the input signal, at least over the time step used for integration. This assumption allows the implicit solution of ζ and ω from knowledge of the past values of input and output to the primary aircraft model. Since there are two unknown variables, data is required from two sets of past values to calculate one set of coefficients. The resulting equations are shown in Fig. 4. It can be shown that this method approaches an exact solution as the frequency of the input signal decreases to zero, assuming that some perturbations still exist to drive the response logic. In a manned flight simulator, perturbations are very sporadic, with considerable gaps of low frequency input. Therefore, this method is well suited for manned simulator applications.

$$\omega = \pm \sqrt{\frac{(c_4 - c_1)}{c_3} - \frac{(c_2 c_{10})}{(c_3 c_9)}}$$

$$\zeta = \frac{c_{10}}{2 \omega c_9}$$

$$\text{where } c_9 = \frac{c_2}{c_3} - \frac{c_6}{c_7}$$

$$c_{10} = -\frac{(c_1 - c_4)}{c_3} + \frac{(c_5 - c_8)}{c_7}$$

$$\begin{aligned} \text{and } c_1 &= \phi''_{i-1} & c_5 &= \phi''_{i-2} \\ c_2 &= \phi'_{i-1} & c_6 &= \phi'_{i-2} \\ c_3 &= \phi_{i-1} & c_7 &= \phi_{i-2} \\ c_4 &= \delta \phi'_{i-1} & c_8 &= \delta \phi'_{i-2} \end{aligned}$$

assuming aircraft behavior of the form:

$$\phi'' + 2 \zeta \omega \phi' + \omega^2 \phi = \delta \phi''$$

Figure 4. Calculation of Coefficients

During some instances this method of calculating ζ and ω can become quite noisy. Therefore, the past 20 values are averaged to compute a smoother signal. For increased accuracy in other applications, a solution using the Fast Fourier Transform or Kalman Fil-

ter may be considered. The penalty will be greater computational time and complexity.

When comparing the approach taken to derive this transfer function extrapolator with other extrapolators it is interesting to note that few other methods consider the full dynamics of the system being modeled. Popular methods such as Euler, Adams-Bashforth, and Runge-Kutta all consider aspects of a single signal (to include derivatives of that signal). However, Rolston's sinusoidal predictor¹ does indeed tune the extrapolator to the frequency of the aircraft. Rolston's premise was that polynomial methods were inadequate to model a system that had sinusoidal characteristics. Using a second order transfer function for extrapolation merely continues and refines Rolston's theme. The transfer function approach takes slightly more computational time than Rolston's method, but considers both the system natural frequency and damping. Of course this aspect of the extrapolator restricts it's application to second order systems, such as aircraft simulations.

Note that this approach also presents a method that is fairly robust to variances in the time between input data arrival. The transfer function can be thought of as a "stand alone" reaction model updating with the same frame time as the primary aircraft model. The critical signal is the input signal, since the logic to modify the characteristics of the transfer function (which are dependent on past values of aircraft state) can occur at a casual or even sporadic interval.

Frequency Response

Using classical methods, the response of this second order transform to an undamped sinusoidal input of the form:

$$\alpha(t) = M \cos \omega_f t$$

can be shown to be:

$$\phi = \phi_m \cos (\omega_f t + \psi_f),$$

where ψ_f is defined as the phase angle in radians:

$$\psi_f = -\tan^{-1} (2 \zeta \beta / (1 - \beta^2)), \text{ where}$$

$$\beta = \omega_f / \omega.$$

The magnitude is determined by ϕ_m as follows:

$$\phi_m = M / \sqrt{((k - J\omega_f^2)^2 + (b\omega_f)^2)}, \text{ where}$$

$$k/J = \omega^2$$

$$b/J = 2 \zeta \omega.$$

Therefore, when the damping ζ , and undamped natural frequency ω are known, it is possible to determine the full response of the system, including phase and gain (magnitude) margins. The exact amplitude ratio and phase angle that this extrapolator will yield will be a function of the calculated ζ and ω . The general trends for these values are shown in Figs. 5 and 6 below.

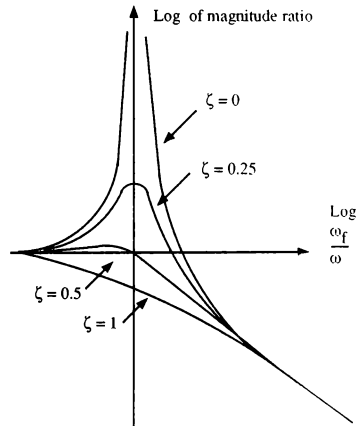


Figure 5. Amplitude Ratio

Since the extrapolator is continually being updated with the latest ζ and ω from the primary aircraft model, it is satisfying to note that it will assume the same frequency response as the primary aircraft

model itself. The solution of the eigenvalues of the characteristic equation of the primary aircraft model will, by default, be the same as those of the extrapolator, assuming an accurate calculation of the system dynamics.

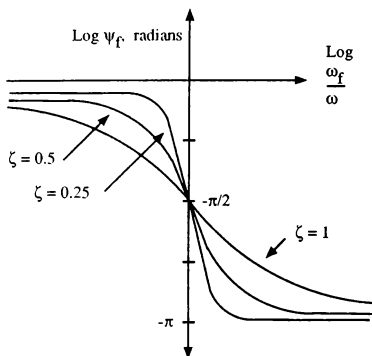


Figure 6. Phase Angle

Off-line Results

To evaluate this extrapolation approach, several off-line and real-time tests were made. Referencing the work of R.E.McFarland², the assumption can reasonably be made that pilot inputs will be in the range of 1-3 Hz, so the test inputs used were 1, 2, and 3 Hz sinusoidal signals of magnitude 1. For parametric evaluations of the effect of change of one variable, an input signal of 2 Hz was used.

As expected, the time response of the extrapolation approach exhibits different characteristics for varying values of the number of iterations (L), the time step factor (T), and the feedback gain (F). To obtain the desired values for a typical 60 Hz iteration, several empirical evaluations were conducted. Fig. 7 shows the effect of modifying L while holding T and F to 1, in response to a 2 Hz input signal. It is interesting to note that iterating the extrapolation more than 3 or 4 times will cause a significant distortion in the shape of the response. It is felt that this is due to the compounding of small errors that may be occur-

ring in the extrapolated response. The value of 2 was chosen for further evaluation.

The effect of modifying the time factor with L and F = 1 can be seen in Fig. 8. It can be seen that modification of the time step used in integration of the transfer function provides an effective way to gain some extrapolation effects, with distortion increasing significantly above T = 5. For further evaluations, the value of 4 was selected.

Since the feedback gain (F) is meaningless unless the number of iterations is greater than 1, the next study used the optimum solution for the previous evaluations. Fig. 9 shows the effect of the change in feedback gain with L = 2 and T = 4. The extrapolated response seems to be improved only slightly for significant changes in this value. This is due to using a small number of iterations. For this specific case, the value of F = 6 was chosen.

The responses of this extrapolator configured with L = 2, T = 4, and F = 6 to a 1, 2, and 3 Hz input signal are shown in Figs. 10, 11, and 12. The transfer function approach is compared with a common second order Euler extrapolation for the same signal. Even though the Euler method is commonly used in flight simulation extrapolations, it can be shown that the transfer function approach produces significantly better results for all input frequencies tested.

It should be noted that other combinations of these values could also produce similar results. Note that the time response is optimized for inputs signals of 2 Hz, which is in the middle of the frequency range to be analyzed. Of particular interest in these graphs is the lack of significant phase lag or magnitude degradation.

Real-Time Results

A test was conducted with an existing in house simulation to evaluate this extrapolator in real time. The simulation was a high fidelity handling qualities model which had several subsections which executed in parallel across several processors. The crew station hardware was read asynchronously at 80 Hz, the primary aircraft model was updated at 240 Hz, the flight controls at 80 Hz, and the visual system output at 60 Hz. The output to the visual system was synchronized, reducing the uncertainty of the transport delay to one half (on the average) of the 80 Hz asynchronous hardware reads, or approximately 6 milliseconds. The extrapolation was done from within the

host computer, just before the 60 Hz output to the visual system.

For the extrapolator test, a signal generator was used to drive the lateral stick directly while simultaneously being recorded by an oscilloscope. The intensity of the visual system blue channel was modified to be directly proportional to the aircraft roll position. The visual system blue signal magnitude was then fed directly into the same oscilloscope, and the delay measurement was made by comparing the phase shift of the two signals. This setup eliminated any extra delay that might have been added by the test equipment. It is important to note that this was not a transport delay measurement of the facility, but rather a real time test of the extrapolator. Therefore, it was important that the aircraft model and flight control system be part of the measurement.

Tests were conducted around the 1, 2, and 3 Hz input signal range. Since the blue signal is shown as a series of dots rather than a clear line, there is some small error in the results due to inaccuracies in pinpointing the peak. Overall, however, the data was remarkably consistent, and the results reported are near the nominal conditions.

The aircraft roll rate time constant from stick input to roll rate response for this simulation was 250 milliseconds. Since the measurand was roll position instead of roll rate, there was an additional 90 degrees of phase lag due to signal integration. This additional phase lag produced an extra time delay that was a function of the input frequency, as noted below:

input signal frequency	1 Hz	2 Hz	3 Hz
roll dynamics	250	250	250
integration delay	250	125	83
total "true" delay	500	375	333
visual system delay	70	70	70
expected measured delay	570	440	403
delay w/ extrap. off	574	439	390

The baseline response to a 1 Hz input signal and the visual response are shown in Fig. 13. Note that cursors are put on the chart to identify the magnitude peaks from which the phase measurements were taken. The upper right hand corner shows the total phase difference between cursors, which was 574 milliseconds for this condition. The 1 Hz response with the extrapolation turned on is shown in Fig. 14.

Note that the extrapolator reduced the 1 Hz delay by 62 milliseconds.

Likewise the response to a 2 Hz input signal is shown in Figs. 15 and 16 with the extrapolators off and on respectively. In this test range there is a benefit of 38 milliseconds due to the extrapolation. The baseline and extrapolated response to a 3 Hz signal is shown in Fig. 17 and 18. Here the extrapolation benefit was approximately 20 milliseconds. The measured extrapolator benefit as a function of input signal frequency is summarized below:

input signal frequency	1 Hz	2 Hz	3 Hz
delay w/ extrap. off	574	439	390
delay w/ extrap. on	512	401	371
extrapolation benefit	62	38	19

In addition, the simulation was flown manually by an engineer who commented that the extrapolated roll response seemed smoother, making it easier for him to complete a roll intensive series of maneuvers. Additional work is planned to obtain qualitative data using a series of high gain man-in-the-loop tests such as mid-air refueling and tracking tasks. These tests will be conducted using operational and former Air Force and Navy pilots.

Conclusion

The implementation of a second order transfer function to extrapolate aircraft state can be shown to reduce the system transport delay significantly across a reasonable range of pilot input frequencies. The extrapolator is fairly insensitive to variances in the arrival time of aircraft state information, making it ideal for networked simulations, including long distance multi-ship simulations. Tuning of the extrapolator can be accomplished, if desired, by iterating several times with feedback. In many situations this extrapolator can reduce the simulator transport delay to half of it's original value, significantly reducing the allowances that simulation users need to make for the digital nature of the aircraft model.

To allow the extrapolator to properly configure itself to different aircraft models, the aircraft damping and natural frequency can be calculated continuously from the past values of the aircraft state using several methods, including one proposed in this paper. Finally, the arrangement of this extrapolator lends itself to coprocessing in parallel with the primary aircraft

model, thereby reducing the transport delay further by one frame.

In these respects, this transfer function extrapolator approach not only effectively compensates for visual system delay, but it also solves many practical implementation issues facing the current generation of flight simulators.

References

- (1) D.R. Rolston, "Sinusoidal Integration for Simulation of Second Order Systems," AIAA Flight Simulation Technologies Conference Collection of Technical Papers, Niagara Falls, NY, June 1983, pp. 52-63.
- (2) R.E. McFarland, "Transport Delay Compensation for Computer-Generated Imagery Systems," NASA TM 100084, Jan. 1988.

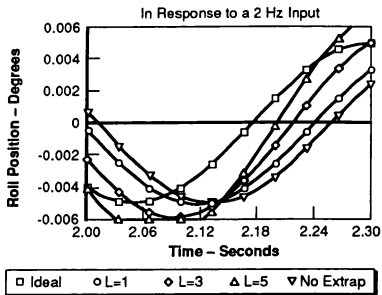


Figure 7. Effect of No. Iterations

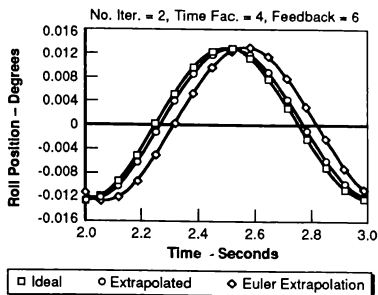


Figure 10. Final Response to 1 Hz Input

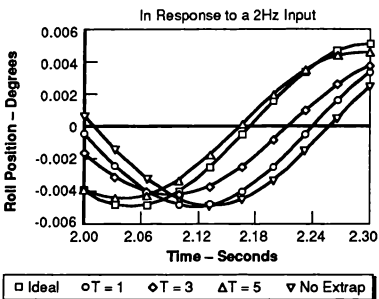


Figure 8. Effect of Time Factor

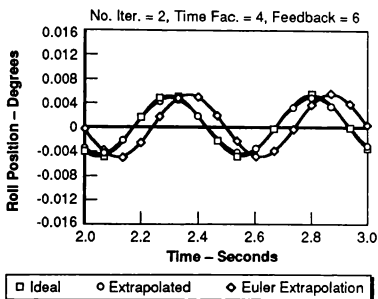


Figure 11. Final Response to 2 Hz Input

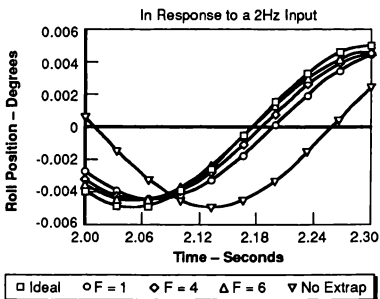


Figure 9. Effect of Feedback Gain

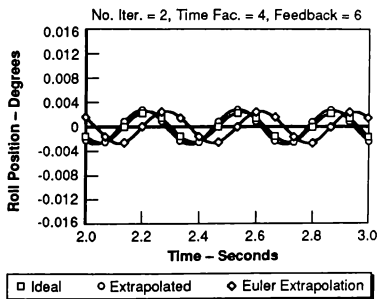


Figure 12. Final Response to 3 Hz Input

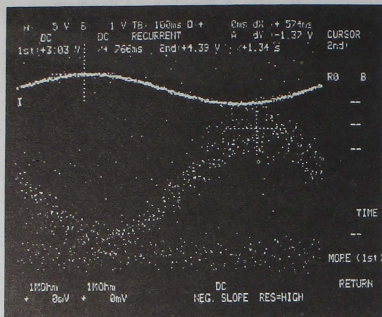


Figure 13. 1 Hz, Extrap Off

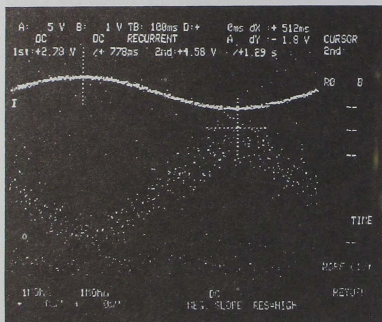


Figure 14. 1 Hz, Extrap On

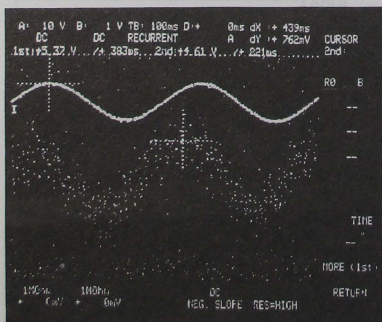


Figure 15. 2 Hz, Extrap Off

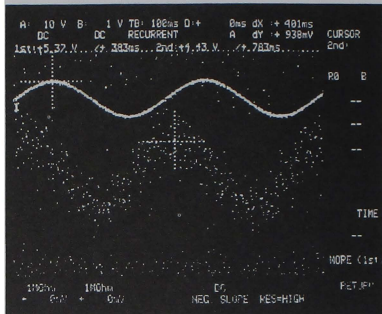


Figure 16. 2 Hz, Extrap On

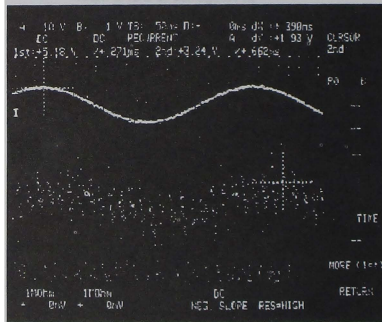


Figure 17. 3 Hz, Extrap Off

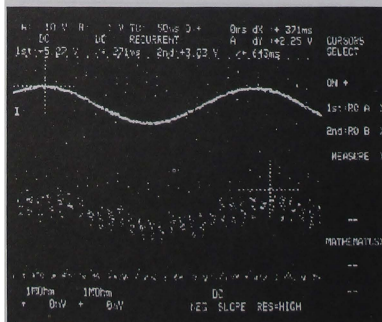


Figure 18. 3 Hz, Extrap On

Aircraft Threat Modeling from Performance Data

Mark R. Anderson*
Virginia Polytechnic Institute and State University, Blacksburg, VA 24061
and

Daniel E. Schab#
Naval Training Systems Center, Orlando, FL 32826

Abstract

A procedure is developed to construct simplified dynamic aircraft models for use as threats in air combat training simulators. Performance data in the form of excess power and energy maneuverability curves are used to define the lift and drag characteristics of the models. A parameter identification method is used to extract model parameters directly from the performance curves. The method therefore insures that the threat model performance matches that of the actual threat. Simplified threat models of two modern fighter aircraft are used to demonstrate the modeling procedure.

Nomenclature

C	aerodynamic coefficient
D	drag force
K	drag polar quadratic
L	lift force
M	Mach number, pitching moment derivative
P _s	excess power
S	wing reference area, matrix of singular values
T	thrust
T ₀	static sea level thrust
U	left singular vectors
V	total velocity, right singular vectors
W	vehicle weight
X	matrix of independent variables
a	speed of sound
b	spline weighting coefficient
g	gravity constant
m	vehicle mass, number of independent parameters
n	load factor
p	roll rate, parameter vector
q	pitch rate
q̄	dynamic pressure
r	yaw rate
s	singular value
u	forward velocity
w	vertical velocity
x	range position
y	cross range position, dependent variable vector
z	height position

α	angle-of-attack
β	sideslip
δ	density ratio (ρ/ρ_{sl})
ϕ	roll angle
η	throttle
θ	pitch angle
ρ	atmospheric density at altitude
ρ_{sl}	atmospheric density at sea level
σ	spline knot location
τ	lag time constant
ξ	spline basis function
ω	turn rate
ψ	heading angle
ω_{sp}	effective short period natural frequency
ζ_{sp}	effective short period damping ratio
$\tau\theta_2$	effective pitch numerator time constant

Introduction

In most modern air combat training simulations, dozens of air vehicle threats are being flown simultaneously in order to mimic modern combat environments. The aircraft threat models must be very simple so that a minimum amount of computational power is used during real-time simulations. In addition to simplicity, the threat models must also be realistic so that the training exercise is representative of actual combat conditions.

Most aircraft threat models in use today have been developed by piecing together manufacturer's data into a simplified model of the threat. For example, Heffley and Mních have developed a method to build "minimum complexity" simulation math models for helicopters.^[1] These helicopter models were not originally intended as training simulator threats; however, the modeling method was developed to reduce computational power during simulation. Heffley's approach uses data from sources such as: flight manuals, manufacturer's stability and control data, Jane's All-the-World Aircraft,^[2] DATCOM,^[3] and military standards.^[4] The modeling method requires only forty-four parameters of a simplified helicopter model to be selected based on the data collected about the specific helicopter to be modeled. Validation of the simplified model consists of comparisons of trim points and time histories generated from an independent high fidelity simulation program. Parameters of the model are adjusted manually to improve validation results, if necessary.

Unlike helicopters, data may be available for aircraft threats in the form of performance charts. Excess power contours and energy maneuverability plots are frequently estimated for threat aircraft to make

* Assistant Professor, Department of Aerospace and Ocean Engineering, Senior Member AIAA.

Aerospace Engineer, Fighter Attack Branch, Member AIAA.

This paper is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

background material for the military flying qualities standard contains considerable data for ω_{sp} , ξ_{sp} , n/α , and τ_R .

Table 1 Threat Model Parameter List

Model Coefficient	Number of Parameters
weight, W	1
reference area, S	1
static thrust, T_0	1
short period natural frequency, ω_{sp}	1
short period damping, ξ_{sp}	1
load factor/angle-of-attack, n/α	1
roll mode time constant, τ_R	1
maximum roll rate, p_{max}	2
maximum Mach number, M_{max}	3
load factor limits, n_{max} , n_{min}	2
engine time constant, τ_{eng}	1
maximum lift coefficient, C_{Lmax}	3
drag polar, $C_{D,0} + KC_L^2$	2
drag coefficient error function, $f(C_L, M)$	variable
total number of parameters	20+variable

Drag Coefficient Identification

The drag force coefficient function $C_D(C_L, M)$ is identified from performance data regarding the actual threat aircraft. Identification of the drag model insures that the *performance* of the threat model closely matches that of the actual aircraft. The identification method currently being considered is based upon principal components regression.^[10-12]

Consider the system of equations represented by,

$$y = Xp \quad (26)$$

where, y is a vector of dependent variables, X is a matrix of independent variables, and p is a vector of unknown parameters. The vector y will consist of N sample points of the dependent variable, in this case, the drag coefficient of the threat. The matrix X will consist of m basis functions which are evaluated at each of the N data points. The parameter vector p , which consists of one parameter value for each basis function, is unknown.

The least squares estimate of the parameters in the vector p is obtained by minimizing the cost function,^[13]

$$J = (y - Xp)^T(y - Xp) \quad (27)$$

which results in the following least squares solution, denoted p_{LS} ,

$$p_{LS} = (X^T X)^{-1} X^T y \quad (28)$$

For the problem of identifying a drag coefficient function from performance data, it is possible that the matrix $X^T X$ is ill-conditioned. If the matrix $X^T X$ is poorly conditioned or rank deficient, the inverse

$(X^T X)^{-1}$ will lead to inaccurate parameter estimates. To circumvent the matrix inversion problem, a solution based on the singular value decomposition of X is utilized. Let the singular value decomposition of X be defined as,

$$X = USV^T \quad (29)$$

where U and V are orthogonal matrices ($U^T U = I$ and $V^T V = I$) and S is a real, diagonal matrix which contains the singular values of the matrix X . Also assume that the singular values are arranged in order of descending magnitude, or $S = \text{diag}[s_1, s_2, \dots, s_m]$ where $s_1 \geq s_2 \geq \dots \geq s_m$. The squared singular values are sometimes called the principal components.^[10]

Now suppose that the singular value decomposition of X is subdivided so that all of the relatively "large" singular values are in S_1 and all of the "small" singular values are in S_2 . The singular value decomposition of X is then,

$$X = [U_1 \ U_2] \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \quad (30)$$

The division of S into two parts, such that $S = \text{diag}[S_1, S_2]$, is carried out defining a singular value threshold.^[10-12] All singular values greater than the threshold are placed in S_1 and all singular values less than the threshold are placed into S_2 . It is fairly straightforward to show that the least squares solution (28) becomes,

$$p_{LS} = V_1 S_1^{-1} U_1^T y + V_2 S_2^{-1} U_2^T y \quad (31)$$

where (30) represents the singular value decomposition of X .

Equation (31) above reveals the effect of the "small" singular values of X which have been included in S_2 . If X is rank deficient, some of its singular values will be very small. When S_2 includes singular values near zero, (31) shows that the second term of the least squares solution can blow up!

At least one parameter identification technique involves replacing the second term of (31) with "a priori" parameter data.^[10] For the threat modeling application, the least squares identification technique will be used to estimate an error function wherein the a priori value of the parameters will be zero. The second term of (31) will be ignored in the parameter estimates because of possible ill-conditioning; therefore, the final parameter values will be given by the following equation,

$$p = V_1 S_1^{-1} U_1^T y \quad (32)$$

To apply the identification technique to the problem of finding an appropriate threat model drag coefficient function, level straight and turning flight excess power contour plots are electronically scanned and digitized to yield data points of Mach number (M),

altitude (h), and excess power (P_s). The excess power data is then converted to an equivalent lift and drag coefficient form using the following expressions,

$$C_D = \frac{2(W/S)}{\rho(Ma)^2} \left(\frac{\delta T_o}{W} \cdot \frac{P_s}{Ma} \right) \quad (33)$$

$$C_L = \frac{2n(W/S)}{\rho(Ma)^2} \quad (34)$$

where a is the speed of sound, ρ is the atmospheric density and W/S is the aircraft's wing loading. The load factor is obtained from the following expression,

$$n = \sqrt{1 + \left(\frac{V}{g}\right)^2 \omega^2} \quad (35)$$

where ω is the aircraft's turn rate. When ω is zero, the aircraft is in straight and level flight and the load factor n is equal to unity. When ω is not zero, the aircraft is turning and the load factor is greater than unity. The turning flight information will help to populate the Mach number, lift and drag coefficient data set, particularly at the higher values of lift coefficient.

The parameter identification scheme is formulated assuming N sample points of Mach number, lift and drag coefficient are obtained from (33)-(35). The drag coefficient is then modeled by the following function,

$$C_D = C_{D,0} + KC_L^2 + f(C_L, M) \quad (36)$$

The drag polar parameters $C_{D,0}$ and K are identified by first assuming that $f(C_L, M) = 0$. The normal equation (26) becomes,

$$\begin{bmatrix} C_{D(1)} \\ C_{D(2)} \\ \vdots \\ C_{D(N)} \end{bmatrix} = \begin{bmatrix} 1 & C_{L(1)}^2 \\ 1 & C_{L(2)}^2 \\ \vdots & \vdots \\ 1 & C_{L(N)}^2 \end{bmatrix} \begin{bmatrix} C_{D,0} \\ K \end{bmatrix} \quad (37)$$

with obvious definitions for y , X , and p . The parameters $C_{D,0}$ and K are identified using the standard least squares solution (28) because conditioning is generally not a problem when identifying the drag polar.

Once $C_{D,0}$ and K have been identified, the nonlinear error function $f(C_L, M)$ is identified using the equation,

$$C_D - C_{D,0} - KC_L^2 = f(C_L, M) \quad (38)$$

In other words, a second identification step is used to identify $f(C_L, M)$ only.

The error function $f(C_L, M)$ is currently represented in terms of a weighted sum of cubic spline basis functions,

$$f(C_L, M) = \sum_{i=1}^{n_c+2} \sum_{j=1}^{n_m+2} b_{ij} \xi_i(C_L) \xi_j(M) \quad (39)$$

with $\xi_i(\cdot)$ representing a single spline basis function and n_c and n_m are the number of spline nodes or "knots" in C_L and Mach number, respectively. The individual spline functions, with k knots located over the interval $[\sigma_1, \sigma_2, \dots, \sigma_k]$, are defined as in Reference [12],

$$\xi_i(x) = \begin{cases} 0 & \text{for } \sigma_1 \leq x \leq \sigma_{i-3}; 5 \leq i \leq k+2 \\ \varphi_1(\eta_{i-3}) & \text{for } \sigma_{i-3} \leq x \leq \sigma_{i-2}; 4 \leq i \leq k+2 \\ \varphi_2(\eta_{i-2}) & \text{for } \sigma_{i-2} \leq x \leq \sigma_{i-1}; 3 \leq i \leq k+1 \\ \varphi_2(1-\eta_{i-1}) & \text{for } \sigma_{i-1} \leq x \leq \sigma_i; 2 \leq i \leq k \\ \varphi_1(1-\eta_i) & \text{for } \sigma_i \leq x \leq \sigma_{i+1}; 1 \leq i \leq k-1 \\ 0 & \text{for } \sigma_{i+1} \leq x \leq \sigma_k; 1 \leq i \leq k-2 \end{cases} \quad (40)$$

with,

$$\begin{aligned} \varphi_1(\eta) &= 0.25\eta^3 \\ \varphi_2(\eta) &= 1 - 0.75(1+\eta)(1-\eta)^2 \\ \eta_i &= (x - \sigma_i)/(\sigma_{i+1} - \sigma_i) \end{aligned} \quad (41)$$

The parameters which are identified from performance data are the spline coefficients b_{ij} in (39). The number of spline function coefficients depends upon the number of spline knots used. From (39), one can see that there will be $(n_c+2) \times (n_m+2)$ spline function coefficients.

It should be clear that the spline function representation allows $f(C_L, M)$ to take on many forms. Its purpose is to model any differences between a simple drag polar and the drag coefficient function needed to match the performance of the actual threat. In addition, the flexibility offered by the spline function representation allows for corrections to be made which stem from the perhaps overly simple thrust model described previously.

There are basically two design choices that must be made prior to identifying the spline function coefficients. First, the spline knot locations in C_L and Mach number must be selected. The spline knots generally should be chosen wherever the drag function undergoes significant variations. For example, it is often helpful to specify several tightly spaced knots near Mach = 1 in order to correctly model drag rise.

The singular value threshold should be chosen as small as possible to yield good model fidelity. However, a threshold set too small will result in a drag coefficient function which is unrealistically complex due to ill-conditioning. Figure 1 shows a set of example identified drag functions wherein the singular value threshold has been varied. Note that much of the function detail is lost when the singular value threshold is chosen as a large value. When the singular value threshold is very large, the total drag function will only contain the simple drag polar (i.e. $f(C_L, M) \approx 0$). As the singular value threshold is reduced, more for the true features of the drag function become evident. However, when the threshold is set too low, numerical ill-conditioning causes the function to attempt to approximate characteristics that are not well represented in the data. Note that the maximum values of the last drag function in Figure 1 (threshold = 0.0001) have been clipped for plotting, but the function is clearly

excessively complicated as a result of numerical ill-conditioning.

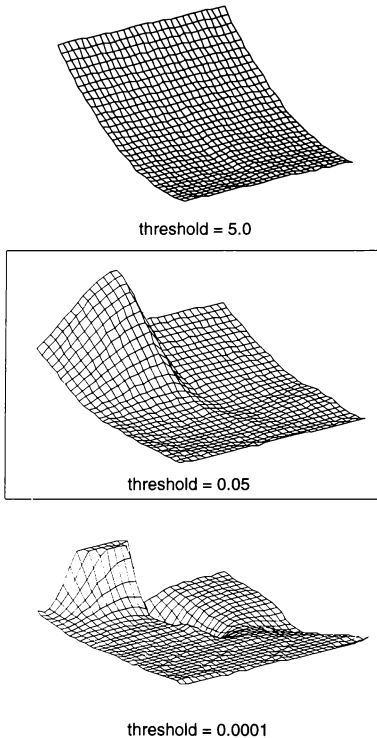


Figure 1 Singular Value Threshold Comparison

Aircraft 'A' Threat Model Example

Threat models for two modern fighter aircraft have been developed using the new modeling concepts. The first example is a typical U.S. fighter. With the exception of the drag and lift function parameters, values for each coefficient listed in Table 1 were selected for Aircraft A from Jane's and the appendices of the military flying qualities standard. When using the military flying qualities standard, Aircraft A was considered to be Level I, Class IV (highly maneuverable) and in a Category A flight condition (air-to-air combat).

Figures 2 and 3 respectively depict the level flight and turning flight excess power contours for Aircraft A. The stars in these figures are the excess power points representing the actual aircraft. These 155 data points are used to identify the aircraft drag coefficient function and were obtained by digitizing the actual performance curves of the threat aircraft.

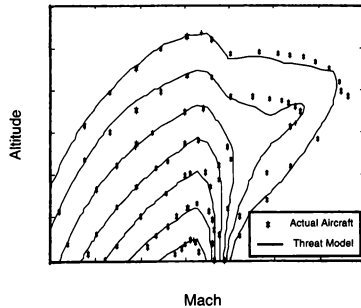


Figure 2 Aircraft A Level Flight Excess Power

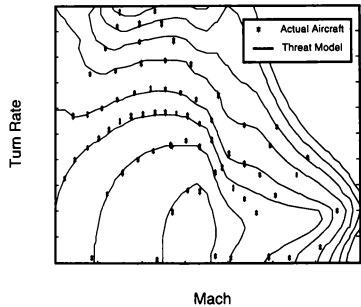


Figure 3 Aircraft A Turning Flight Excess Power

The solid lines in Figures 2 and 3 are the excess power contours which are predicted from the new threat model. Figure 2 reveals that the threat model level flight performance closely matches the actual aircraft. Figure 3 shows that the threat model matches the turning performance of actual aircraft very well in the subsonic region, but the match is not as good in the supersonic region. Part of the matching problem is due to the fact that only 4 (out of 155) data points lie in the high Mach number and low turn rate region.

Figure 4 illustrates the identified drag coefficient function obtained for Aircraft A. The spline knot locations for Aircraft A were chosen as: [0.2 0.8 0.9 1.0 1.2 1.6] for Mach number and [0.0 0.2 0.8 1.4 2.0 3.0]

for lift coefficient. The singular value threshold was chosen as 0.05. With 6 knots in Mach number and 6 knots in lift coefficient, there are 64 parameters needed to model the cubic spline error function depicted in Figure 4. As a result, 84 total parameters are being used to model Aircraft A.

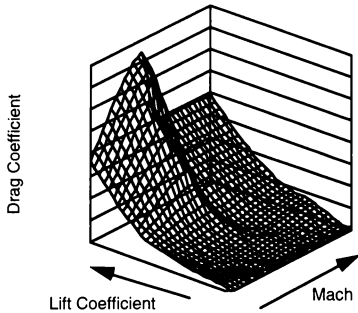


Figure 4 Aircraft A Drag Coefficient Function

As an additional validation of Aircraft A, the response of the threat model was compared to a very high fidelity simulation of the actual aircraft. Seven different maneuvers were performed at four flight conditions throughout the flight envelope of the aircraft. The maneuvers included level accelerations and decelerations, bank-to-bank rolls, longitudinal doublets, and wind-up turns. For these preliminary tests, both the threat model and the high fidelity simulation were flown using open-loop "batch" maneuvering commands formulated from ramp and step functions. More extensive closed-loop maneuvering tests are planned.

Figure 5 shows a sample comparison of the open-loop tests. The time responses shown in Figure 5 stem from a 10,000 ft altitude level acceleration starting at Mach = 0.4 and ending Mach = 0.7. These responses demonstrate that the threat model behaves in much the same manner as the actual aircraft (represented by the high fidelity simulation) when flying through several different excess power levels.

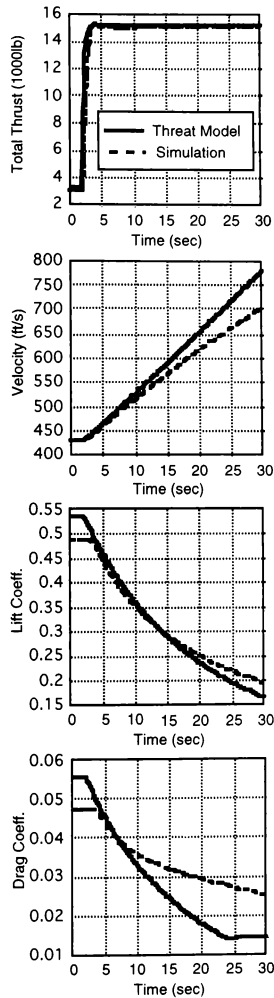


Figure 5 Aircraft A Level Acceleration Time Response

Aircraft 'B' Threat Model Example

A second modern fighter aircraft threat model has been developed to further exercise the new modeling concepts. In contrast to Aircraft 'A', no manufacturer's data or higher fidelity simulation model is available for the foreign-built Aircraft 'B'. Therefore, Aircraft 'B' represents a more typical application of the threat modeling methods wherein very little quantitative data is available other than some performance estimates.

Figure 6 illustrates the identified drag coefficient function obtained for Aircraft B. A total of 94 data points were used to identify the aircraft drag coefficient function. The spline knot locations for Aircraft B were chosen as: [0.2 0.4 0.9 1.0 2.1 1.1] for Mach number and [0.0 0.2 0.4 0.5 0.8 1.1] for lift coefficient. The singular value threshold was chosen as 0.1. With 6 knots in Mach number and 6 knots in lift coefficient, there are 64 parameters needed to model the cubic spline function depicted in Figure 6. Aircraft B, like Aircraft A, therefore requires 84 modeling parameters.

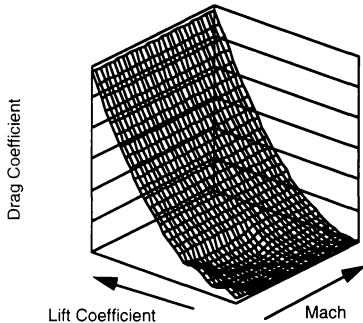


Figure 6 Aircraft B Drag Coefficient Function

Figure 7 depicts the level flight excess power contours estimated for Aircraft B and the new threat model developed from these estimates. The stars in Figure 7 are the excess power points representing the actual aircraft while the solid lines are the excess power contours predicted from the new Aircraft B threat model. After completing the identification process, the results shown in Figure 7 reveal that the new threat model of Aircraft B closely matches the level flight performance of the actual aircraft threat.

The process of developing the new threat model is fairly simple and has resulted in a level of performance fidelity ordinarily not possible using more traditional methods. In addition to evaluating the overall fidelity of the new model, a comparison can be made with an existing threat model of Aircraft B. The level flight excess power contours of the existing model are shown in Figure 8 and represent the limited threat model fidelity typically found in use today. Note that much of

the detail in the high Mach number range is missing from the existing model (Figure 8) as compared to the new model (solid line in Figure 7). Therefore, the new model contains significant level flight performance fidelity improvements at high Mach numbers.

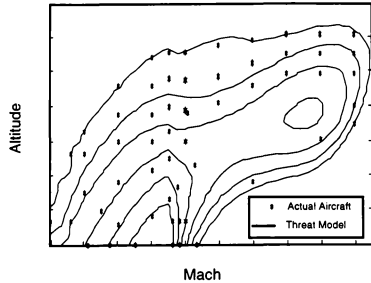


Figure 7 Aircraft B Level Flight Excess Power

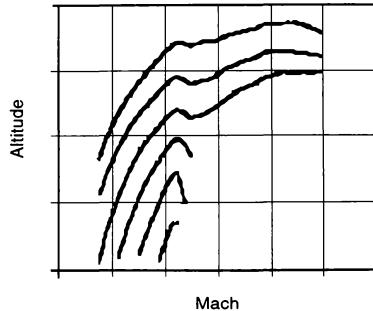


Figure 8 Previous Aircraft B Model Excess Power

However, the success of the outcome is very dependent on the number of modeling parameters used and the location of the drag function spline knots. For example, 67 data points were selected for matching the level flight excess power contours. By comparison, only 27 data points of the aircraft's turn performance curves were used for matching. Figure 9 shows that the resulting drag coefficient function leads to a relatively poor match between the model and the actual aircraft maneuvering performance. Based on the results from both aircraft model examples, it is expected that an increase in turn performance data points would also lead to an improvement in turn performance fidelity.

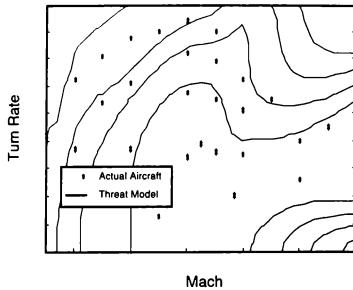


Figure 9 Aircraft B Turning Flight Excess Power

Conclusions

The example results shown in this paper reveal that surprisingly detailed drag coefficient functions can be extracted directly from typical aircraft performance charts. This capability has been used to form a systematic procedure for aircraft threat modeling. The performance of the resulting threat model closely matches that of the actual threat aircraft. The parameter identification method used in the modeling procedure has only a limited number of design choices; therefore, the level of engineering expertise required and the time needed to develop new aircraft threat models should be significantly reduced. Finally, threat models created using the new model structure can be easily modified or upgraded whenever new performance estimates are obtained.

Acknowledgments

The research reported in this paper was sponsored by the Naval Training Systems Center under Contract N61339-92-C-0100 and the Naval Air Systems Command. The authors would like to thank Mr. R. Thomas Galloway for helping to define the research objectives. The authors are also grateful to Brent York, Chris Weekley, and David Honaker for their help in developing the threat modeling software.

References

- [1] Heffley, R.K. and Mnich, M.A., "Minimum-Complexity Helicopter Simulation Math Model," NASA CR-177476, April 1988.
- [2] Taylor, J.W.R., ed., *Jane's All the World's Aircraft*, Jane's Publishing Inc., New York, 1983-1984.
- [3] *USAF Stability and Control Datcom*, Air Force Flight Dynamics Laboratory, WPAFB, OH, Jan. 1975.

- [4] "Flying Qualities of Piloted Aircraft," MIL-STD-1797, March 1987.
- [5] Etkin, B., *Dynamics of Atmospheric Flight*, John Wiley & Sons Inc., New York, 1972.
- [6] Burgin, G.H., et.al., "An Adaptive Maneuvering Logic Computer Program for the Simulation of One-on-One Air-to-Air Combat, NASA CR-2582, Sep. 1975.
- [7] Kelley, H.J., "Reduced-Order Modeling in Aircraft Mission Analysis," *AIAA Journal*, Vol. 9, No. 2, Feb. 1971, pp. 349-350.
- [8] Williams, D.H. and Wells, D.C., "Traffic Scenario Generation Technique for Piloted Simulation Studies," NASA TM-86397, April 1987.
- [9] Anderson, J.D., *Introduction to Flight*, McGraw-Hill Book Company, New York, 1989.
- [10] Anderson, L.C., "Robust Parameter Identification for Nonlinear Systems Using a Principal Components Regression Algorithm," AIAA Paper No. 85-1766, Atmospheric Flight Mechanics Conference, Snowmass, CO, Aug. 1985.
- [11] Anderson, M.R., "Pilot Model Structure Identification Using Principal Components Regression," AIAA Paper No. 91-2893, Atmospheric Flight Mechanics Conference, New Orleans, LA, Aug. 1991.
- [12] Cloutier, J.R., "Multivariate, Minimum-Curvature Splines for Randomly-Space Data," AIAA Paper No. 91-2744, Guidance, Navigation, and Control Conference, New Orleans, LA, Aug. 1991.
- [13] Strang, G., *Linear Algebra and its Applications*, Academic Press, New York, 1980.

D³S: THE DELPHINS DISPLAY DESIGN SYSTEM

E. Theunissen*
Delft University of Technology
Delft, The Netherlands

Abstract

Development of software for dedicated display systems can be a laborious and time-consuming task. This makes it difficult to benefit from the enormous flexibility which is offered by programmable display systems. Various tools exist which aid the designer with the development of display formats. For high-fidelity simulation, powerful and hence expensive graphics workstations are required. At the Delft University of Technology, a display design system has been developed which can be used for high-fidelity simulation of displays at only a fraction of the cost of workstation based systems.

The Delphins Display Design System (D³S) incorporates a new method for the design of aircraft instrumentation displays which allows the specification of display formats and their dynamic properties in a structured, easily understandable, unambiguous and hardware-independent way. From this specification, software for the dedicated display systems is automatically generated. D³S is currently being used at the Delft University of Technology and the National Aerospace Laboratory NLR.

Introduction

In the cockpit of today's modern commercial aircraft, electronic displays are more and more replacing the conventional mechanical instruments. Due to the enormous flexibility of the programmable display systems, there is a great potential for significant improvements in the information interchange between man and machine.

In order to optimize the information transfer, display formats should present the required information so, that the effort required for the perception, interpretation and evaluation of the information is minimized. In other words, the display should be intuitively understandable.

Because the development of such display often requires multiple evaluation and modification phases, the design and modification of the formats should not require a huge programming effort. Several tools exist which aid the designer with the development of display formats (O'Donnell¹, Montoya²).

In June 1990, DELPHINS (DELt Program for Hybridized Instrumentation and Navigation Systems) was initiated at the Delft University of Technology. The main goal of this program is to develop suitable presentation methods for four-dimensional navigation and guidance information which can be used during the manual and the supervisory control task.

To be able to carry out this research, a display design and evaluation system is needed.

In general, the development of display formats is an iterative process which consists of the following four phases:

1. idea
2. specification
3. implementation
4. evaluation

Most existing display development systems assist the designer during the specification of the concept by means of a CAD program. The implementation and evaluation of the display format can be performed at different levels. Beaden³ distinguishes between the following four:

1. Static formats
2. Animation
3. High fidelity simulation
4. Testing within the vehicle.

This is an approach which is used in a number of existing display design systems. Static format development takes place on a graphics workstation, after which interactive animation is possible via some kind of command interpreter. High fidelity simulation takes place on a powerful graphics workstation which can generate the display formats at an update-rate of 20 to 30Hz. Testing the concept within the vehicle often requires rehosting of the display software to a different target system. Both Honeywell¹⁰ and Sextant Avionique² have automated part of the rehosting process.

A general problem with the rehosting of software is that this often requires lots of programming by hand. This is especially so if the output of the display design system to the code-generator does not contain a description of the dynamic properties of the display format. As a result, the part of the program which actually moves objects around on the screen is often still written by hand. Another problem with display software development is that sometimes modifications are made at the implementation level rather than at the specification level. Therefore one must make sure that in situations where it requires less effort to change the display software than to go back to the design system, the changes are still made through the design system. Otherwise, the specification does not correspond with the final display format.

A solution to both problems is the introduction of a specification language which contains a description of the display format and its dynamic properties. This specification

*PhD student, Electrical Engineering

must be generated by the design system, and functions as the input to the automatic code-generator.

By enabling designers to edit the specification, the situation where changes are made at the implementation level can easily be avoided. During the development of the Delphins Display Design System D³S, special emphasis was placed on the use of such a specification method.

The D³S display design and evaluation system has been developed at the faculty of Electrical Engineering and consists of special hardware and software which allow rapid generation of complex, high-quality display formats. The hardware which is necessary to achieve a realistic simulation of display formats is determined by the resolution and display update-rate criteria. To achieve a low price/performance ratio, a display system which is based on off-the-shelf PC hardware has been selected as the target system.

Currently, the DELPHINS research environment consists of a number of display design and evaluation systems, and a moving-base flight-simulator with an electronic display mounted in front of the right seat for a realistic evaluation of the display formats.

Selection of the display hardware

Because of the rapid innovations in the field of computer graphics systems, the selection of the display hardware posed a difficult task, which took place in september 1990. The following criteria were used during the selection:

1. The system should be capable of achieving a display update-rate of approximately 20Hz.
2. The displays should be large enough to show rather complex pictures which are still readable, but on the other hand they should fit in the flight simulator. Hence the choice was limited to monitors from 12" to 16".
3. Because displays in real aircraft use a hybrid raster/stroke-writing technique resulting in smooth straight lines, the system in the simulator should have a resolution high enough to reduce the staircasing effect to a level where it is hardly noticeable.
4. The system should be able to display different colors at the same time.
5. The system should be able to produce smoothly animated pictures.

Criteria 1,2 and 3 are related to each other in the following way: When the size of a monitor increases, the resolution of the display should also increase in order to maintain the same pixel size, which is set by the spatial resolution of the eye and the viewing distance. Because of the increase in resolution, the performance must also increase to maintain the same update-rate.

As the maximum number of colors is determined by the number of available bitplanes, system performance is also influenced by the number of colors.

Criterion 5 requires the presence of at least two frame-buffers in order to use double-buffering.

Two options were evaluated: PC-based graphics systems and workstation based graphics systems. It was apparent that a PC would be limited by its graphics architecture (CGA/EGA/VGA) both in speed and in resolution. Only a PC with a SVGA display would be able to generate pictures with a high-enough resolution. Due to the relatively slow update-rate of SVGA adapters, the SVGA display would only do for format design and animation. Hence when using a PC, some kind of additional graphics engine would be required for high-fidelity simulation. During the evaluation of workstation based graphics systems, it became apparent that powerful workstations were required in order to achieve a high-enough display update-rate. In contrast to this, current EFIS displays often do not contain more processing power than a conventional PC. It was decided to try to use a 386PC as a target system, with an additional graphics engine to unload the main processor for the drawing operations. A number of graphics engines for PC's were evaluated, after which a board based on the TMS34020 from Texas Instruments was selected. The documentation and development tools which are available for TMS34020 based systems is tremendous, and standardisation is provided by the TIGA⁶ (Texas Instruments Graphics Architecture) definition. Besides the two criteria mentioned previously, this system offers enormous flexibility by allowing the user to create his own graphics primitives and run them on the graphics engine.

The complete display system consists of a 386/387 PC with a VGA monitor, and a TMS34020 Software Development Board with a hi-resolution monitor.

Specification of the display formats

An important aspect of display research is the ability to specify the display format in a structured, straightforward, easily understandable, unambiguous, and hardware independent way. This avoids possible confusion between persons who are involved in the display design and evaluation process, allows efficient generation and use of documentation, and provides the possibility of automatic software generation.

To develop a specification method and a code-generator, a structured analysis of the final product (the display program) is necessary. From this analysis it is determined which elements of the display software must be specified.

A display program produces a display format from a set of dynamically changing inputs and a set of fixed rules. A display program can be divided into a display format specific and a display format independent part.

Figure 1 presents the general structure of a display program.

In this figure, the input data is processed in a tree containing the display logic, clipped and scaled according to a set of rules and then fed to the transform algorithms. These algorithms modify the standard display (zero-input condition) according to a set of predefined rules.

This data is used to build a display list, which contains the commands necessary to generate the desired format. The display list is processed by an algorithm which controls the

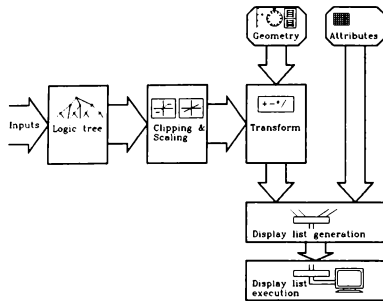


Figure 1 General structure of a display program

display hardware.

Instead of input coming from the 'outside world', data can also come from virtual on-screen devices (e.g. buttons, sliders, knobs, potentiometers). Input than takes place by pointing at a certain location on the screen with a device that outputs x-y coordinates (e.g. a mouse, track-ball, light-pen, or touch-screen). These x-y coordinates are transformed into an output value of the virtual device which can be connected to an input of a virtual instrument.

Because the limiting/scaling and transformation is performed according to a set of rules, these rules must be specified. Furthermore, the geometry and the properties of the display layout must be specified.

A specification method, from now on referred to as DDF⁵ (Display Definition Format) has been developed with which it is possible to specify the information describing the static and dynamic behaviour of a display in a structured, hardware-independent, easily understandable and unambiguous way.

In a DDF specification a display format is divided into a number of independent objects. Each object is subdivided into a number of base-objects. Base objects are the smallest building blocks, and have a number of attributes (color, style, priority, etc.). Each base-object can be translated, rotated and scaled independently of the other ones. If a transform must be applied to all base-objects, it can be performed on the object. Figure 2 shows the data flow between the geometry data area and the display list generation.

The dynamic behaviour of the display is obtained by transforming the geometric data according to the specified set of rules, and the current outputs from the limit and scale section.

As can be seen from Figure 2, transforms are first performed on the base-objects, and thereafter on the complete object. The base-object geometry data passes from the geometry data area to the base-object transform algorithms. After all base-objects have been transformed, the object is constructed, and stored in the object buffer. Next, the object

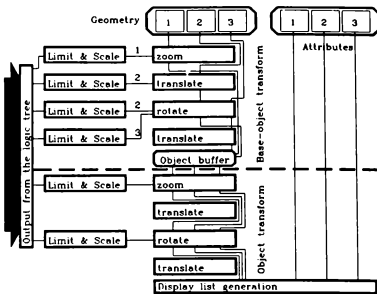


Figure 2 Data flow in display program

transforms can be performed. The resulting object is passed to the display list generation section. Note that the base-objects can have different connections between the base-object transforms, but not between the object transforms. Because each inserted base-object can have a number of transforms attached to it, the specification of each inserted base-object must contain the information about the allowed transforms. Furthermore, the specification of each inserted base-object contains the name of the geometric description of the base-object.

Generation of the specification

A CAD program is used to design the base-objects and the objects. Because the output of the CAD program only contains a description of the static display format, a proprietary tool, from now on referred to as DXF2DDF is used to specify the dynamic properties. DXF2DDF generates the DDF specification, which is plain ASCII text, and easy to understand. The DDF structure reflects the structure and hierarchy of the specified object. Because a display format may contain more than one object, the names of the different objects are stored in the display object list. Figure 3 illustrates the specification process.

From specification to display software

The process of automatic software generation in D³S is based on combining a dedicated set of software modules for the display format independent part with automatically generated modules for the display format specific part (Theunissen⁶). The sourcecode generator converts the hardware independent DDF specification into hardware dependent source code for a certain target system, e.g. the 486/TIGA system. Code generators have been developed which allow rapid rehosting of the display software to other target

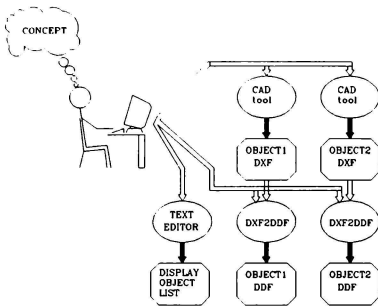


Figure 3 Specification process

systems.

From the information contained in the DDF file, the sourcecode-generator creates the connections between the geometry data area, the base-object transform algorithms, and the object transform algorithms. (Figure 2). Furthermore, the sourcecode generator constructs the limiters and scalars and connects their inputs to the input data section and their outputs to the object and base-object transforms. In this way, the display dynamics can be controlled through the inputs. The resulting sourcecode does not contain the geometric description and the properties of the base-objects. These are loaded from the DDF file during the initialization of the display program. In this way, changes can be made to the geometry and the properties of the base-objects by editing the specification, without having to regenerate the display program.

Some target systems contain special hardware to speed up the generation of a display format. To make optimal use of this hardware, the code-generator contains a rule-base which is used to optimize the software for a specific target system.

The time required for the code-generator to produce sourcecode typically is less than five seconds. The process of generating an executable display program from the DDF specification consists of several steps and typically takes a few minutes. Figure 6 presents an overview of the different actions which must be performed. Sourcecode must be generated for all the objects in the display. A test program must be generated which allows the user to modify each of the specified inputs to the display. Next, all sourcecode must be compiled and linked with a number of libraries.

To aid the designer, a shell has been built around the code-generator, the compiler and the linker, which allows the user to produce an executable display program from the specification files by typing a single command.

Results

To test the system, students from the faculty of Aerospace Engineering have used D³S to design instruments for their specific application. A number of typical aircraft instruments and control panels have been implemented, among which are a so-called Primary Flight Display (Figure 4) and an EFIS Control Panel (Figure 5).

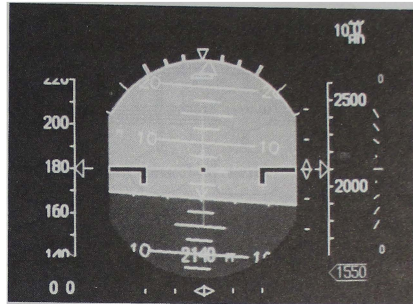


Figure 4 PFD generated with D³S

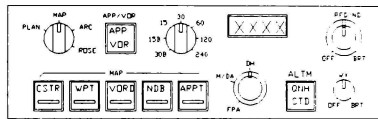


Figure 5 EFISCP generated with D³S

It has been demonstrated that the complete design of a typical aircraft instrument display from scratch to an animated version can be performed within a few hours.

After the animation stage, high-fidelity simulation is performed on a TIGA PC which can be connected to the flight-simulation computer. On the TIGA display system typical aircraft displays (e.g. a Primary Flight Display) run at an update rate of 20 or 30Hz. Update rates of 60Hz can be achieved for more simple display formats.

For a realistic evaluation of the display formats, a 14" CRT has been installed in the right instrument panel of the moving-base flight simulator at the faculty of Aerospace Engineering of Delft University. A number of experiments have been conducted with this system. (Theunissen⁷, Verhoeven⁹, Visser¹¹). Furthermore, the system is used to develop the display software for the student display stations in the recently acquired Cessna Citation II aircraft.

Besides conventional aircraft instruments, new types have been developed with D³S. One example is a so-called 'Tunnel-in-the-Sky' display' which is being used to provide the pilot with integrated four-dimensional guidance and navigation information.

The fact that the specification is easy to understand and can be modified with a simple text editor has proven to be valuable. First of all, changes to the display can be performed within a few minutes, and because most of them do not even require regeneration of the source code, recompilation is not necessary. Experienced users often require only a few seconds before a modification is implemented.

This significantly reduces the time which is required for the iterative evaluation/modification phase. Furthermore, the use of the specification allows easy exchange of specific display objects between different display formats. In this way, a library of display elements can be built. Finally, because all changes are made at the specification level and not at the implementation level, the specification always corresponds with the display format. By using code-generators there is no apparent difference whether software is generated for animated format evaluation, high-fidelity simulation, or in-flight testing.

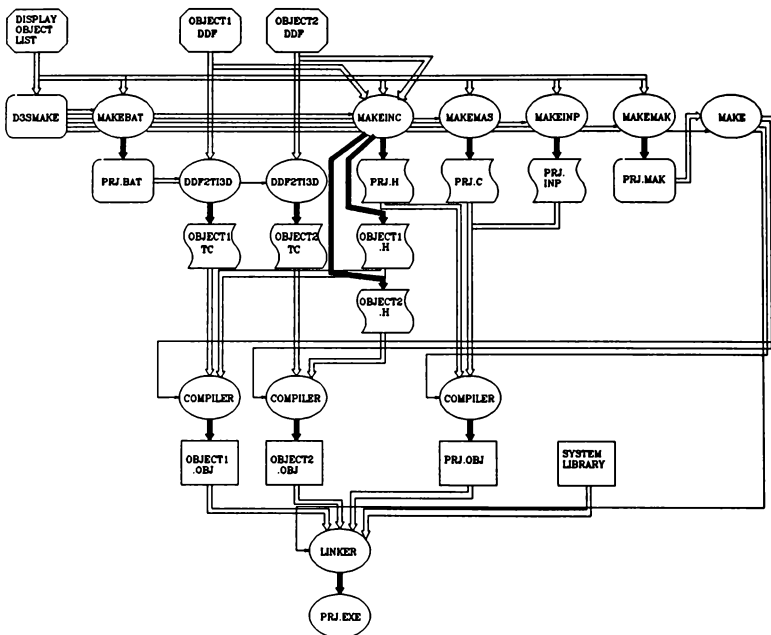


Figure 6 Overview of the software generation

Conclusion

With the Delphins Display Design System, prototyping and animation of display formats can be performed efficiently on conventional PCs. The cost required for high-fidelity simulation have been drastically reduced by using PC hardware combined with an additional graphics engine.

D²S allows designers with no knowledge about the specific graphics hardware to develop complex display formats such as a Primary Flight Display in a relatively short time.

The introduction of a hardware independent description of the display format and its dynamic properties makes it possible to automatically generate software for an animated display on different target systems.

The possibility to change the geometry of the base-objects without having to regenerate and recompile the source-code contributes significantly to the flexibility of the system, because minor changes to the display format can be performed within a few minutes.

Total development time reduces seriously due to the more efficient modification/evaluation phase, and the easy progress from animated evaluation to in-flight testing.

References

- [1] Beadon, A.A. (1988). Requirements for Rapid Prototyping of Crew Station Displays. SAE Aerotech, pp.151-155.
- [2] Durgeat, E. and Debort, P. Benefits of formalization and automatic coding in the development of embedded software. Sextant Avionique, France.
- [3] Montoya, R.J., Turner, T.L., Jewell, D.M., Aanstoos, J.V., Suresh, R., Barker, M.C. (1991). AGSSS: The Airborne Graphics Software Support System; An Ada/PHIGS-Based Display Editor for the Rapid Development of Cockpit Display Software Systems. WL-TR-91-1042, Wright Laboratory, OHIO.
- [4] O'Donnel, C.M., Smith, M.W. (1989). The Interactive Display Design Tool: An Application Program for Part-Task Simulation Development. Proceedings of the Human Factors Society 33rd Annual Meeting. pp.375-379.
- [5] Theunissen, E. (1991). DDF: The Display Definition Format. Master's Thesis, Delft University of Technology, Faculty of Aerospace Engineering, Department of Stability and Control.
- [6] Theunissen, E. (1991). D²S: The DELPHINS Display Design System. Master's Thesis, Delft University of Technology, Faculty of Electrical Engineering, Department of Telecommunication and Traffic Control Systems.
- [7] Theunissen, E. (1993). The DELPHINS Tunnel-in-the-Sky program. Delft University of Technology, Faculty of Electrical Engineering. - to be published -
- [8] TIGA Interface User's Guide (1990). 2564001-9721 revision B. Texas Instruments Computer Video Products.
- [9] Verhoeven, N.J.V. (1993). De actieve en passieve side-stick en vliegtuig-besturingscomputers in een realistische simulatie. Master's Thesis, Delft University of Technology, Faculty of Aerospace Engineering, Department of Stability and Control.
- [10] Virtual Prototypes (1993). With VAPS, Honeywell Creates a Full Cockpit Display in 1 Month. INTERFACES, Volume 2, Number 1.
- [11] Visser, B. (1993). Waarnemings-onderzoek naar snelheidspresentatie op EFIS displays. Master's Thesis, Delft University of Technology, Faculty of Aerospace Engineering, Department of Stability and Control.

A High Fidelity Video Delivery System For Real-Time Flight Simulation Research

Daniel A. Wilkins & Carl C. Roach

*SYRE, a division of SYSCON Services, Inc.
NASA Ames Research Center
Moffett Field, CA.*

Abstract

The Flight Systems and Simulation Research Laboratory (Simlab) at the NASA Ames Research Center, utilizes an extensive network of video image generation, delivery, processing, and display systems coupled with a large amplitude Vertical Motion Simulator (VMS) to provide a high fidelity visual environment for flight simulation research.

This paper will explore the capabilities of the current Simlab video distribution system architecture with a view toward technical solutions implemented to resolve a variety of video interface, switching, and distribution issues common to many simulation facilities. Technical discussions include a modular approach to a video switching and distribution system capable of supporting both coax and fiber optic video signal transmission, video scan conversion and processing techniques for lab observation and recording, adaptation of image generation and display system video interfaces to industry standards, an all raster solution for "glass cockpit" configurations encompassing Head-up, Head-down, and Out-the-Window display systems.

Introduction

The Simlab complex currently supports three research facilities. The Interchangeable Cab, Advanced Cab, and the Vertical Motion Simulator laboratories. The Interchangeable Cab laboratory is utilized as a staging area for interchangeable cab buildup, test, and integration before transition to the Vertical Motion Simulator. The Interchangeable Cab laboratory also serves as a full-up, fixed base simulation facility with dual cockpit capability. The Advanced Cab laboratory supports dual cockpit bays in a similar capacity as the Interchangeable Cab laboratory including a 20 foot dome simulation cockpit. The Vertical Motion Simulator laboratory operates an interchangeable cockpit on a large amplitude, six degree-of-freedom, motion system. Due to the flexible architecture of Simlab flight simulation resources all three lab facilities can, and often do, operate simultaneously. (Figure 1)

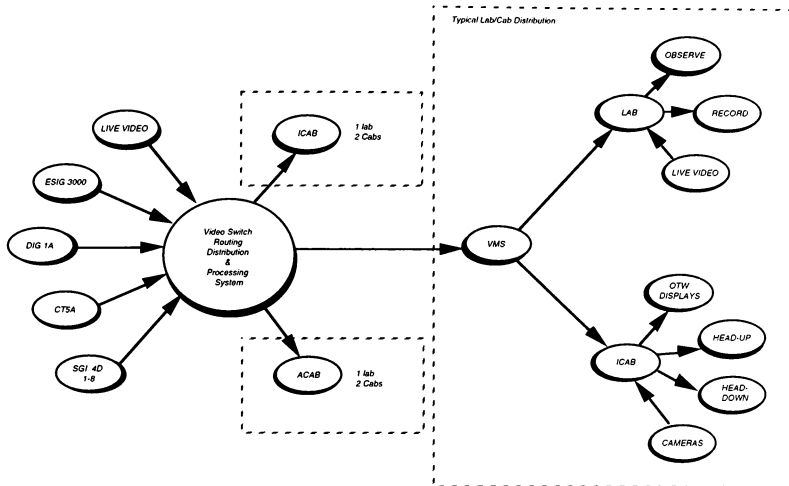
Simlab utilizes a complement of four interchangeable cabs (ICABs) for flight simulation research cockpits. Each ICAB is designed around a typical class of aircraft. For example, R-CAB and N-CAB cockpits provide a four window wide field-of-view configuration of three forward and one chin window for helicopter and vertical short take-off and landing (V/STOL) simulation research. F-CAB features three contiguous windows characteristic of a fighter configuration and S-CAB provides a three window configuration for side-by-side transport aircraft such as the Space Shuttle. Each ICAB supports a variety of instrument and controls systems to provide maximum flexibility of cockpit configuration. ICAB windows use a wide-angle-collimating (WAC) mirror/beamsplitter optical system coupled with a 25 inch (diagonal) high resolution color display monitor to provide out-the-window (OTW) visual flight cues. ICABs also support several Head-up display (HUD) cockpit graphics systems including a 25° x 30° Flight Dynamics Inc. holographic HUD, a 16° x 20° FOV dual combiner Kaiser HUD, and a Honeywell 30° x 40° Integrated Helmet Display Sighting System (IHADSS). Head-down display (HDD) systems include three 8" and three 14" full color, high resolution CRT based display systems.

Real-time, out-the-window, visual flight scenarios are provided by a 4 channel vertical raster format Singer-Link DIG1, a 3 channel programmable raster format Evans & Sutherland CT5A, or a 6 channel programmable format Evans & Sutherland ESIG 3000¹ computer image generation system. Head-Up/Down cockpit graphics are provided by a cluster of eight Silicon Graphics Inc. (SGI) 4D series workstations.

The image generation systems at Simlab are shared resources. Video output from any IG can be made available to any one or all of the research laboratory/cab facilities on demand. For example, a simulation can be started at a fixed base location then relocated to the VMS for motion based operation. Consequently, scene fidelity and visual cue

¹ Scheduled for service 1/94

Figure 1. Simlab Video Delivery System Block Diagram



repeatability are of utmost importance at all locations.

Video Switching and Distribution

The video switch matrix is the central collection and distribution point for all video resources at Simlab. The primary purpose of the switch matrix is to provide seamless routing of video resources to multiple research laboratories and flight simulation cockpit display systems. This is accomplished using a modular architecture of $N \times M$ crosspoint video switch matrices where N is the number of inputs and M is the number of outputs. Matrices are grouped physically and functionally to provide the desired source/destination interface (Figure 2).

Currently, Simlabs video routing switcher supports three basic configurations: a 30 input by 60 output, 35 MHz, RGB matrix to support out-the-window visuals, a 30 input by 40 output, 35 MHz, single line matrix to support lab observation and recording, and a 16 input by 30 output, 100 MHz, RGB matrix to support HUD and HDD requirements. In addition, the 16 x 30 matrix is capable of supporting both fiber optic and coax transmission

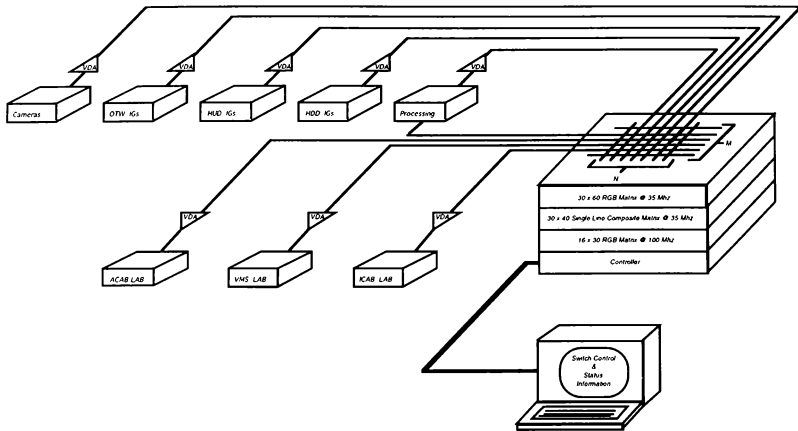
cable interfaces.

Each switch matrix is monitored and controlled from a central PC based system controller. The switch controller provides single or batch mode video switch routing, status information, RS232 interface for local or remote access, and software control of matrix configuration. The video switching system also houses video processing equipment required to provide NTSC video data for lab observation and recording of simulation flight scenarios.

At Simlab, each of the research lab/cab facilities is located more than 500 feet from the image generation systems. In addition, there is a 200 foot flexible cable catenary between the VMS ICAB motion platform and the VMS laboratory. Typically, IG video output drivers are designed to transmit video signals approximately 150 feet at best. Even at these relatively short distances video signal integrity can be compromised.

To compensate for longer distances Simlab employs a system of video distribution amplifiers (VDAs) to maintain video signal integrity between each source and destination. VDAs can be used as buffer/receivers or as long line drivers.

Figure 2: N x M Video Routing Switch Matrix and Distribution System Block Diagram



Buffer/receiver VDAs are configured for AC coupled differential operation. AC coupling is used throughout the distribution system to eliminate the need to maintain a DC reference over long transmission distances and through multiple levels of switching. Differential input amplifiers provide common mode rejection to minimize cyclical interference such as 60 and 400 Hz typically found in the simulation environment. VDAs are also used to buffer and distribute inputs to the video routing switch matrices.

When used as long line drivers, VDAs are capable of driving up to 500 feet of 75Ω video coax cable. VDA drivers provide equalization and phase compensation to minimize frequency loss and phase variation characteristic of long line coax transmission media. VDA output drivers can be configured for AC or DC coupling. AC coupling is also used for driver configuration to minimize potential interference coupling onto the video signal. DC restoration of the video signal is accomplished within the display system thus minimizing loss of the DC characteristics of the video information during transmission.

Recently, Simlab has integrated a routing/distribution system upgrade which utilizes a modular architecture of distribution amplifiers as inputs or outputs to the routing switch matrix.

Distribution amplifier modules can be either coax or fiber optic allowing mix and match transmission of video data via fiber optic or coax cable systems. Fiber optic distribution amplifier modules utilize 850nm infra-red transmitter/receivers to provide up to 100 MHz video transmission bandwidth for up to 0.5km of 50/125 micron fiber optic cable.

Signal bandwidth and transmission distance define the cable type required to interface video resources. At Simlab, 75Ω video coax (RG 59/U or equivalent) is used for distances <100 feet and signal bandwidth < 20MHz. 75Ω video coax (Belden 8281 or equivalent) is used for distances up to 500 feet and signal bandwidth up to 35MHz, and 50/125 micron multimode fiber optic cable is used for distances up to 0.5km and signal bandwidth over 35MHz. Fiber optic cable is also used in applications where the transmission medium is subject to significant ground potential differences, EMI, or RFI signal interference.

Observation and Recording

The ability to monitor and record visual flight cues and pilot response is a basic requirement of flight simulation research. At Simlab, video observation and recording requirements vary with each simulation project. Video monitoring requirements can be as simple as reproducing a single

out-the-window view or as complex as duplicating the full compliment of cockpit visual cues including HUD symbology overlaid onto the background scene and multiple HDD displays.

A centralized network of video processing equipment was developed to provide a flexible architecture of video resources capable of supporting multiple research laboratories under a wide variety of observation and recording situations. The system is comprised of video scan converters, mixers, digital effects generators, a quad splitter, master sync generator, and a variety of test equipment. (Figure 3)

Incorporating the video processing system into the video routing switch has several advantages. For example, genlock synchronization of video effects processing equipment is simplified, timing delays due to cable length variations are minimized, and local switch routing of video resources provide optimum configurability of processing resources. A master sync generator is used to provide genlock synchronization of all video processing systems in order to maintain image stability and color integrity throughout the entire video processing system.

Scan conversion is used to translate the high resolution video output of an image generation system to NTSC format for video processing, lab observation, and video tape recording. Currently, Simlab employs three video scan conversion systems. These include a camera/monitor system, platform specific circuit boards, and programmable digital scan converters.

Camera/monitor scan converters are used to translate the vertical scan video format of the DIG1 IG to standard NTSC video. A high resolution, studio quality camera is focused onto the face of a high resolution display monitor driven by the DIG1 IG. The camera output signal (which is in NTSC video format) is genlocked (synchronized to a master sync source) to provide timing compatibility with video processing systems. This method is only used for the DIG1 due to it's unique vertical scan raster format. There are minor anomalies associated with this technique. There are two diagonal beat lines that scroll through the picture due to the timing difference between the camera and IG vertical sync rates. In addition, some of the picture is lost due to aspect ratio cropping caused by positioning the camera 90 degrees off axis from the display field-of-view. This technique is also used when the CT5A is operated in the DIG1 emulation mode.

Digital scan conversion is another method used to convert high resolution IG video outputs to NTSC format. There are two types of digital conversion units currently in use at Simlab. The first

is a stand alone system that provides programmable video format conversion over a wide range of scan rates. Programmable digital scan converters are capable of adapting to a variety of video input formats. These units also provide programmability of the scan converted video output format and storage of pre-programmed input/output parameters. Programmable scan conversion systems are primarily used to generate conformal IG/HUD video mix.

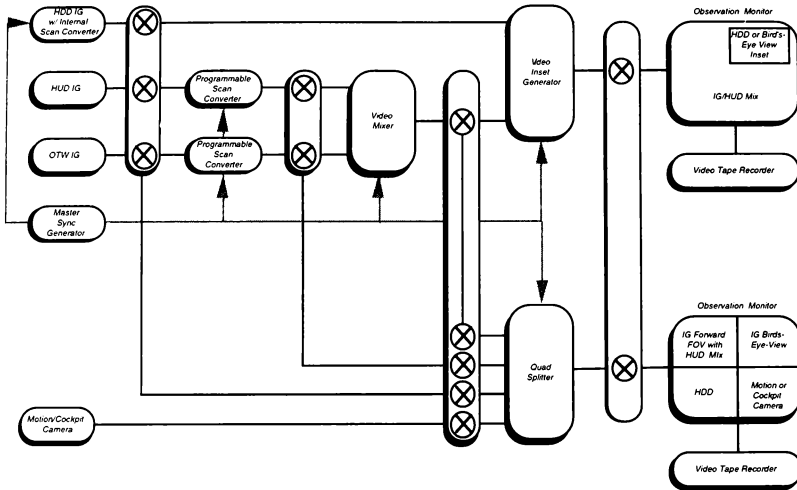
The second type of digital scan converter is a platform specific internal circuit card that performs the video scan rate conversion within the IG. This type of scan converter is typically incorporated in the SGI graphic workstations used to generate HUD and HDD images. It is important to note that the internal scan conversion circuit boards do not provide programmability of the horizontal and vertical signal characteristics of the scan converted video output. Consequently, they cannot be used for conformal IG/HUD video mixing. In addition, the on-board scan converter option does not support interlaced display mode operation of the SGI graphics workstation. Each high resolution image generation system video output is scan converted to NTSC video format for processing, observation, and recording.

Video mixers combine the outputs of two scan converters into a single image with one scan converted image serving as the background and the other as an overlay onto the background. This configuration is used to generate the IG/HUD mix video effect with the scan converted IG image as the background and the scan converted HUD symbology as the overlay. It is essential that the scan converted images are genlocked to a master sync source in order to maintain stability of the image mix. In most cases conformal tracking of the HUD symbology onto the background scene is required.

At Simlab, the OTW IG systems provide a 4:3 image aspect ratio and the HUD IG systems generate a 4:5 image aspect ratio. In addition, HUD symbology normally covers only a small portion of the OTW field of view. Programmable digital scan converters provide image scaling and position control required to produce a conformal IG/HUD image mix. Conformal IG/HUD image mix is a standard feature for lab observation and recording at Simlab.

Digital video effects generators combine the outputs of two scan converters into a single image. One scan converted image serves as the background and the other is a reduced full scale image inset onto the background display. This process is used to provide a bird's-eye-view or a HDD image inset onto the out-the-window IG/HUD mix image. A quad-splitter provides the capability to display up to 4

Figure 3: Typical Video Processing Configuration for Lab Observation and Recording



separate scan converted images on a single display. Again it is important to note that genlock synchronization of all video processing systems is required for stable image position and color lock.

Each laboratory has a minimum of two overhead observation monitors and two recording suites. A recording suite consists of a 19" color monitor, a 3/4" Umatic or a 1/2" VHS video tape recorder, and a time/date generator as required. One recording suite is fixed and the other is portable. NTSC formatted video information is provided to the overhead monitors and recording suites via the switch routing matrix and distribution system.

Video processing, lab observation, and recording systems are based on the NTSC video standard to take advantage of the plethora of equipment available to the broadcast industry. The NTSC video standard is an internationally accepted format which provides the research community with a highly portable video record of their work.

IG and Display System Interface

One of the more challenging aspects of integrating multiple IG and display resources via a centralized routing switcher is adapting customized simulation IG and display interfaces to video industry standards. Over the past few years there has been concerted effort to develop common "standardized" ² video interface formats for shared video resources at Simlab. The philosophy behind this effort is to provide the highest possible image quality and still maintain maximum flexibility of the video delivery systems.

One of the most notable changes in the Simlab video delivery system has been conversion of HUD/HDD IG and display formats from calligraphic (stroke) to high resolution raster based systems.

² Current trends toward multimedia video application of computer graphics systems is forcing a closer relationship between video industry standards and computer image generation video formats.

Calligraphic HUD/HDD IG interface consisted of a DC coupled, bipolar (± 5 VDC), video format coupled with specialized calligraphic display systems using a custom designed, DC coupled, point-to-point distribution system. Routing IG output signals to the remote display systems was accomplished using a manual patch bay. Scan conversion required the use of a camera/monitor scan converter system which made conformal IG/HUD video mixing difficult and inconsistent.

State-of-the-art anti-aliasing techniques and high resolution raster technologies made it possible to replace the calligraphic IGs with high performance graphics workstations and still maintain image quality required for HUD symbology. Although the transition from calligraphic to raster based HUD/HDD format required replacement of the IG systems, calligraphic display technology lent itself to direct conversion from stroke to raster. This process entails modification of the deflection system to provide a constant raster sweep at multiple line rates and incorporation of an RGB video interface to modulate beam intensity for pixel emulation. This process can be applied to both monochrome and color CRT based calligraphic display systems. Monochrome calligraphic CRTs do not require a shadow mask. Consequently, the line and point quality of the image is only limited by the pixel density and anti-aliasing capability of the IG.

Conversion of HUD/HDD image generation and display systems from calligraphic to raster based video format resulted in several significant enhancements to the Simlab video network. HUD/HDD graphics can be provided in full color RGB as well as standard monochrome format. Raster based HUD/HDD IG video interface formats can be readily adapted to common video standards simplifying video routing, processing, and distribution. The quality and consistency of HUD/IG video mix has improved due to the use of programmable video scan conversion. HUD/HDD display systems now support multiple line rates and image linearity has improved dramatically.

Another significant improvement to the Simlab video delivery system has been the gradual transition from an IG driven, vertical scan, OTW video format toward video industry standard horizontal raster interface formats. When the video switch router and distribution system was initially installed at Simlab, the only OTW IG system being supported was the Singer-Link DIG1. The DIG1 was designed with a vertical scan raster format to maximize image content in the vertical display axis for the purpose of vertical lift aircraft flight

simulation research. By default, the vertical scan format became the defacto video interface standard at Simlab. Only recently has Simlab been able to take advantage of programmable IG output and multiple line rate display technologies to escape the restrictions of this non-standard OTW video interface.

The transition from vertical to horizontal scan format began with the acquisition and integration of an Evans and Sutherland CT5A image generation system. The display processor subsection of the CT5A was modified to support programmable raster scan video output formats. The CT5A was then programmed to emulate the vertical raster format of the DIG1 and to provide an 875 line standard horizontal raster format. Multiple line rate, horizontal raster format, OTW display systems were specified and purchased to support both the CT5A and SGI image generation systems. Simlab is currently in the process of replacing the DIG1 with an Evans and Sutherland ESIG 3000. The ESIG 3000 design incorporates a microcode programmable video format generator capable of adapting the ESIG video output to match both the CT5A and SGI video interface characteristics.

Other video interface adaptation techniques employed at Simlab include the use of high bandwidth passive attenuators to translate IG video and sync output levels to conform with video industry standards and external sync adders to convert separate RGB&V video display interface format to RGB with sync-on-green video. This minimizes potential video/sync delay induced, visual anomalies and the number of video transmission paths required to support each display system.

Conclusion

The combined capabilities of programmable format image generation systems, multiple line rate displays, and a common interface to a centralized routing switcher provide a high fidelity visual environment capable of supporting multi-display "glass cockpit" visual flight simulation research at the NASA/AMES Flight Systems and Simulation Research Laboratories.

HYBRID COMPLEX OF THE AIRCRAFT INTELLECTUALIZED CONTROL SYSTEMS
SIMULATION AT THE STAGE OF THEIR RESEARCH PROJECTING

A. I. Yakovlev, V. M. Vasilets
International Academy of Engineering
Russia, Moscow

Abstract

In the paper a the problem of aircraft intellectualized control systems simulation at initial development stage, i.e. under the research projecting, is discussed. The hybrid complex, of simulation including both a computer system and investigation facilities, in which the flight factors are reproduced and their influence to the system and a man is examined, is analysed. The main attention is given to the flight factors influence to pilots and operators. It is supposed, that the research projecting is being carried out in conditions of the cooperative dialogue of many specialists and the base that the vector of system efficiency indices is formed.

Simultaneously with a mathematical simulation of flight vehicle functioning operation the participation of pilots in the control makes it possible to estimate the system efficiency, as well as the crew performance as a whole and its separate members.

The tasks optimisation system of the "best" selection system is formulated taking into account the features of human participation.

Serviceability of the method of simulation and system selection is illustrated on example of the comparison of spaceship landing system.

Nomenclature

- (S1,...,Sn) - n-systems, which need to simulate;
(O1,...,Om) - m-pilots and test-subjects, performing the flight vehicle control;
(A1,...,Ar) - r-intellectualized algorithms, realized by the computer of flight vehicle;
 $Y(\cdot)$ - measurable functions and parameters during the simulation;
 $\delta_k \in E$ - underterminated K-th parameter of flight factors given by the set of its values E;
 $G \in \Sigma$ - random P-th parameter of the flight factors, given by the set of its values;
 $H\Phi(\cdot)$ - technical indices of efficiency;
 $[C_i, \bar{C}_i]$ - allowances on values of efficiency indices and system quality, $i=1,...,m$;
t - current time of system operation;

- $P(\cdot)$ - probability;
 $H_M(\cdot)$ - vector of biomedical indices of the operator performance;
 $H\Phi(\cdot)$ - vector of ergonomic indices;
 $Hc(\cdot)$ - vector of subjective indices;
 $\theta_{j\mu}$ - according coefficient upon the rate;
 $\Omega_{j\mu}$ - domain of rate acceptable values.

The intellectualized control systems
of flight vehicles

The increase of control efficiency at the cost of realization of artificial intelligence methods is highly actual direction in the current development stage of the different flight vehicles: aircrafts, aerospace systems, orbiters. It was specifically result from the occurrence of new technologies in the information processing, an enhancing man-computer interface under the decision of different conventional tasks, the development of adaptive and expert systems, opening the new capabilities of human operation in the extremal conditions.

Let us consider the generalised intellectualized control system (Figure 1).

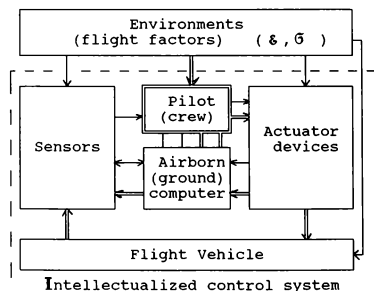


Fig.1. Generalized scheme of intellectualized control system of the flight vehicle.

The primary elements of this system are: the pilot or a crew of flight vehicle, (under the control of spaceship movement it can be a shift change on duty in the Mission control centre); on-board ground computer complex, information sensors, actuator devices, flight vehicle and environment (flight factors).

The pilot participation in the control makes the considering system of

intellectualized one as the man creatively realizes the control functions, gives an estimation of concrete situation and only on the basis of these values puts into effect the control. The traditional control systems contain the relations between elements (shown a single line). At the same time in the advanced generation of computer systems the new relations (they are shown by double line) become available, which are produced as a result of the intellectualized computer operation: by way of formation on the pilot display of critical fashions of the control, required information on the indicators, as well an adaptive effect to the control elements and full-pressure suit depending on the flight conditions. By way of example of such a computer operation one can show the adaptive pressurisation system of flight full-pressure suit under an aircraft movement control during prolonged manoeuvring (Figure 2).

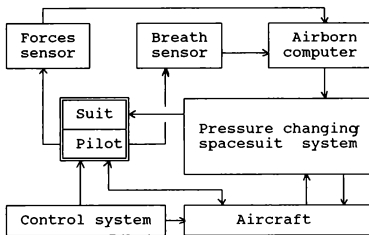


Fig.2. Adaptive pressurisation system of flight.

For formalization of the research designing of the flight vehicle (FV) we are accepting, that it is necessary to simulate the final set of systems

$$(S_1, S_2, \dots, S_n) \quad (1)$$

It is clear, that in the control the final quantity of pilots and test-subjects can be involved

$$(O_1, O_2, \dots, O_m) \quad (2)$$

Let us suggest, that FV-computer can realize r-intellectualized algorithms

$$(A_1, A_2, \dots, A_r) \quad (3)$$

Then all measurable parameters, characterizing its quality and attainment of the operation aim, will be estimated by the function in the form

$$V(t) = Y(t; S_i; O_j; A_y) \quad (4)$$

It is necessary to consider, that the function (4) depends on not only from the meanings (1) - (3) but also from random r and interminate flight factors, i.e.

$$Y(t) = Y(t; S_i; O_j; A_y; \xi; G) \quad (5)$$

For efficiency performance of the complex in interests of several specialists it is necessary to realize the cooperative dialogue. The different hybrid modelling

complexes of intellectualized control systems simulation are suitable for obtaining of the function meanings. The simulation complex for the control system optimization can be seen in Fig.3. It allows to simulate the influences to system and an operator-test-subject.

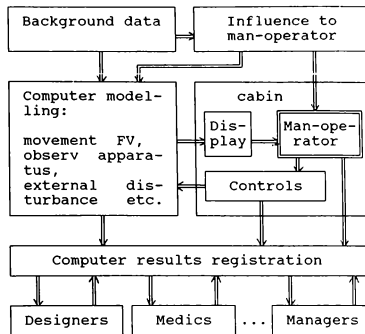


Fig.3. The hybrid simulation complex.

As in the Earth conditions it is not always successful in forming the flight factors effects, to the system and man, then the hybrid complexes, which are located on the flight vehicle, have received the propagation.

For more simple presentation we shall consider, that the errors of measurements of system parameters under the simulation include in the composition of random and indetermined influences as the protective means against this type of influences the filtration methods are successfully used. Also we shall add to the external effects such important processes, as the change of performance and state of man as a result of the system elements influence.

For an example the sensors, located directly on the body surface, cause the irritation of man under prolonged experiments.

The quantity valuation and efficiency of the control systems are determined by the system of indices, which are used upon the simulation for evaluation of the system, human performance and mission efficiency, carried out a flight vehicle. Hence let us analyse in more detail the set of indices, used under modelling of the man-machine intellectualized control systems.

The set of indices for the man-machine system analyses

The functioning of engineering and man-machine systems during the flight is a

process, in which an action of each system is specified of its goal and intention. In the theory of efficiency the purposeful operation of the system, in which there were determined the goals, influencing factors and the means of aim's achievement, is defined conceptually as the term operation. The use of the operation concept and its models allows quantitatively to define the efficiency indices that characterize the degree and extent system achievement of useful effect, as well as the time expenditure and different reserves. Quantitative meanings of the efficiency indices are formed both indirectly through the altered measured meanings under the modelling, and on the basis of mathematical relations utilization, in which the qualitative indices of system and its elements are the argument. The second approach under the man-machine system investigations essentially shortens the time of research designing.

Upon the investigation of efficiency of the man-machine control systems and its quality the next indices have been obtained an extensive distribution: economies in labor and finances, a system application, a reduction of operating personnel, provision of comfort conditions for the human operation, simplification of methods, forms, techniques and means of the information reception, needed for system operation, release of an operating personnel from performance of routine operations, increasing of system operation efficiency in extreme conditions, provision of the personnel operational safety, decreasing of system negative influence to the ecological medium.

Upon the investigation of efficiency of the man-machine control systems and its quality there are useful not only technical H^t , biomedical H_m , ergonomic H_0 indices, but the users subjective values of system functional efficiency and the operators working comfort H_c .

Depending on the system application the different indices are included into a vector rate of the system efficiency and quality in the form of separated components.

The technical indices of the efficiency as a whole represent

$$H^t = (H^t_1, H^t_2, \dots, H^t_n), \quad (6)$$

which for instance has the next elements: system operation H , the accuracy of received decisions H^t etc.

Under the investigation of the control systems a wide recognition gained the system index $P(\cdot)$ - a probability if inoutput of system technical indices H^t, \dots, H^t_n from the given allowance $[C_1, \dots, C_n]$, which is a function of more simple indices and its acceptable meanings in certain time moment t

$$P(t; S_i; O_j; A_k; g; \sigma) = P\{C_1 < H^t < \bar{C}_1, \dots, C_n < H^t < \bar{C}_n, g \in E, \sigma \in \Sigma\}, \quad (7)$$

For the quality valuation of human performance under the control analysis of convenience in human operation, provision of a comfort integration with computer there are used the biomedical indices of human status during the operation, which form a vector

$$H_m = (H_m^1, H_m^2, \dots, H_m^r), \quad (8)$$

where H_m^1 - average number of heart beats, H_m^2 - average number of breathings etc.

For reception of human biomedical indices the next electrophysiological indices are used: bioelectrical cortical activity, i.e. encephalogram - $\Psi_1(t)$, bioelectrical muscular activity, i.e. electromyogram - $\Psi_2(t)$, galvanic skin reaction - $\Psi_3(t)$, electrical indices of human heart activity etc.

The capability of an identification of the different operators states with help of the index (8) is confirmed by significant investigation experience of control operator performance. For example the recorded results of electrophysiological indices under modelling of the control systems of flight vehicles, are seen in Table 1.

Table 1.

Biomedical indices of operator performance			
Indices	Encephalogram components and recording conditions	Stick control	Push-button control
Average indices of bioelectrical cortical activity integrated units	Delta-one and theta rhythm 1 - 8 Hz	33,5	34,3
Average rate of galvanic-skin reaction on Tarhanov, mc V	before operation during operation after operation	580 906 770	560 730 380
Average rate of heart beats, b/min	before operation during control after operation	78 89 80	78 82 80

The Table 1 shows, that the human discrete control formation with the help of push-button device against the making of the same operation with the help of the contact stick causes the stress to a lesser extent.

The important indices of operator performance efficiency are ergonomic indices

$$H_0 = (H_0^1, H_0^2, \dots, H_0^n) \quad (9)$$

may have the next elements: H_0^1 - mathematical expectation of a pilot response lag under usage of control action, H_0^2 - quantity of control movements etc.

The opportunity and appropriation of the ergonomic indices utilization are proved on an example of comparative valuation of operator working results with the pointer and digital indicators (Table 2).

Table 2.

Operators	Ergonomic indices	Pointer indicator	Digital indicator
1-st	average rate of wrong readings, % average time of readings understanding, s	15,9 7,1	0,6 0,1
2-nd	average rate of wrong readings, % average time of readings understanding, s	16,7 5,9	0,7 0,1

The analysis of ergonomic indices, represented in Table 2, shows, that the operators are better working with the digital indicator, as they spend less time on the control action realization and make less the wrong readings. It will be noted, that the used in this case the ergonomic indices are informative for an operators comparison under their work with the used pointer and digital indicators.

From represented instances of the analysis of pilot operating activity (Tables 1,2) it is necessary to use the instruments and its adequate patterns, with which the man had operated, but a data processing must make on the computer. It is that determines the need for production of the hybrid modelling complex.

It is commonly used under the analysis of man-machine systems of flight vehicles the pilot subjective values, which form a vector of subjective indices

$$H_c = (H_c^1, H_c^2, \dots, H_c^K) \quad (10)$$

one from its elements the well known Cooper scale may be.

For a complete estimation of the system and the operator performance under modelling it is necessary to set up the integrated index, including as elements the indices $H_T, H_x, H\phi, H\delta, i.e.$

$$\begin{aligned} H(\cdot) &= \{ H_y^\mu(\cdot) \}, \\ \mu &= [T, \phi, M, C], \\ y &= [1, \dots, n]. \end{aligned} \quad (11)$$

The index (11) more completely describes the working efficiency of the man-machine systems in the operation. However the use of vector index (11) makes difficult an interpretation of received data and a comparison of different patterns of the system.

That is why in the engineering practice the resultants of indices are useful.

For instance the index is formed as linear sum of vector elements, i.e.

$$H_x = \sum_{\mu} \sum_{y} \theta_{\mu y} H_y^\mu, \quad (12)$$

where - agreeing coefficients. Another commonly used index is probability of inoutput of partial indices from the domain of admissible learnings (meanings) Ω_y in a certain time moment "t", i.e.

$$P(t) = P[H_y^\mu(t) \in \Omega_y]. \quad (13)$$

The index (6)...(13) are a function from measured during the modelling and therefore for (11), as well as for others indices, it is true

$$H[Y(t; S_1, O_1, A_x; \delta, \zeta)] = \{ H_y^\mu(\cdot) \}. \quad (14)$$

The conceptual importance under the formation of indices has that fact, that an assembly of indices must be total and all specialists, participating in the research designing, should have the informative indices for a comparison of alternate system patterns.

During the simulation process of complicated systems a contradiction occurs: the more completely the index describes the system features and its used efficiency the more difficult to get its quantitative expressions. The means overcoming this contradiction are the use of computer or computer networks with high computing capabilities.

It will be noted, that selection of the system functioning efficiency index and its quality valuation is determined in many cases by subjective decisions of the investigators, as the formal methods, permitting on the basis of system investigation to determine the required range of local efficiency indices, are absent in the present time. It is especially hardly to solve the selection task of system efficiency index, requiring the consideration of plural nature of influencing factors uncertainties, realized strategies.

On the basis of the simulation results and quality indices analysis of the intellectualized control systems at the final stage of the research designing the "best" system is selected from the set of alternative or significantly less set of the effective system is created. Hence let us analyze system comparison criteria and formulate its possible decision tasks.

The selection tasks of the "best" man-machine system

The simulation of system S for the set of uncertain E and random, operators O and algorithms A in the case of use of the indices such as (13) allows to create matrix

$$[F_{ij}^k(t; \delta, G)] = \begin{bmatrix} P(t; S_1, 0, A_k; \delta, G) \dots P(t; S_1, 0, A_k; \delta, G) \\ P(t; S_m, 0, A_k; \delta, G) \dots P(t; S_m, 0, A_k; \delta, G) \end{bmatrix}. \quad (15)$$

In case, when the vector indices such as (14) are used, that after the simulation realization the assembly of matrices on particular indices become available.

$$[H_{ij}^{\mu k}] = \begin{bmatrix} H_{y1}^{\mu k} & \dots & H_{ym}^{\mu k} \\ \dots & \dots & \dots \\ H_{yn}^{\mu k} & \dots & H_{ym}^{\mu k} \end{bmatrix}. \quad (16)$$

$$x=1, \dots, r; \mu=1, \dots, n; \gamma=T, \exists, M, C.$$

As a follows from (15) the selection task of the "best" system at the research designing stage becomes complicated by the lack of information about the training, state and capabilities of a pilot, which will realize the system control. At the same time from expression (19) follows, that under identification the "best" system all the same difficulties, which occurs under the decision of polycriterial tasks, also are run.

The selection of the "best" intellectualized control system under the decision of monocriterial tasks on the index (13) on the basis of the matrix (15) may be formulated in one from next methods depending on the operators participation in the control.

The task 1

All matrix meanings (15) are thought to be known as a result of the simulation. The "best" operator, i.e. obtaining the most index meaning (13), will take part in the control. Then the "best" systems is interpreted as the systems, which provide for holding of conditions

$$(S_1^*) \in \arg \max_{S_1} \max_{A_k} \max_{O_j} P(t; S_1, 0, A_k; \delta, G), \quad (17)$$

in case of absence of unique solution, and

$$S_1^* = \arg \max_{S_1} \max_{A_k} \max_{O_j} P(t; S_1, 0, A_k; \delta, G), \quad (18)$$

under the unique solution of task, where S_1^* - the optimal system for task solution 1. If the flight vehicle control is realized by crew, when an its each member must guarantee the task decision, that system selection logically to make on the principle of a guarantee result. Then instead of the task 1 we have a next task.

The task 2

Under made assumptions about a crew members operation the "best" system is that, which provides for holding of condition

$$(S_2^*) \in \arg \max_{S_1} \max_{A_k} \min_{O_j} P(t; S_1, 0, A_k; \delta, G), \quad (19)$$

if the unique task solution is absent and

$$S_2^* = \arg \max_{S_1} \max_{A_k} \min_{O_j} P(t; S_1, 0, A_k; \delta, G), \quad (20)$$

if the unique task solution exists, where S_2^* - the optimal system for the task 2 solution.

The optimal systems, obtained from terms (17) and (19) or (20), are extreme or limit, as they restrict the set of its variants. For instance by way of insertion of randomization the other systems may construct. In this case the insertion of random variable under the flight vehicle control by crew is a practicable way of randomization. Then the system selection task is formulated as follows.

The task 3

Under assumption that a single member of crew appointed randomly performs the flight vehicle control, that the "best" system is one, which provides for holding of condition.

$$(S_3^*) \in \arg \max_{S_1} \max_{A_k} \min_{O_j} \sum_{i,k,j} P_{ij}^k \gamma_i^k \psi_j^k, \quad (21)$$

where γ_i^k - realization frequency ratios of k-th algorithm and i-th intellectualized control system, ψ_j^k - control frequency ratio of j-th crew member, (S_3^*) - set of optimal systems S under the solution of task 3. It is clear, that when in a game the saddle points is presented that

$$\max_{S_1} \max_{A_k} \min_{O_j} P_{ij}^k = \min_{S_1} \max_{A_k} \max_{O_j} P_{ij}^k$$

and then $(S_2^*) \in (S_3^*)$, but if the saddle point is alone, that $S_2^* = S_3^*$. In cases, when a crew training is done after the system selection, that it can increase the control efficiency, using the additional information about the system, if in the matrices (15) the saddle points are absent. As in this case it is true that the possibility to get again Π become available, which will equal a next value in the case second task solution

$$\Pi^0 = \min_{O_j} \max_{S_1} \max_{A_k} P_{ij}^k - \max_{S_1} \max_{A_k} \min_{O_j} \sum_{i,k,j} P_{ij}^k \gamma_i^k \psi_j^k. \quad (22)$$

It is clear, that a transfer from the second task to the first one under the system solution makes it possible to get a guarantee increment of the system efficiency increase equal

$$\Pi^1 = \min_{O_j} \max_{S_1} \max_{A_k} P_{ij}^k - \max_{S_1} \max_{A_k} \min_{O_j} \sum_{i,k,j} P_{ij}^k \gamma_i^k \psi_j^k. \quad (23)$$

As it will be seen from formulated tasks, a selection of the "best" system depends on both the crew operation strategy and the randomization insertion crew flight vehicle informing.

The formulated tasks 1...3 are generalized and in cases of vectorial indices of matrices (16). It may be realized, if we introduce the utilization principle of consecutive criteria under a system selection. Such an approach to a problem naturally complicates the rule of systems comparison, as it demands the successive solution of tasks 1...3 on each

index.

However, seeing that under considered approach to the simulation of intellectualized control systems the information on system quality indices and its used efficiency are obtained to the full in the course of single experiment, but the processing of received information is realized in the computer set with the use of cooperative dialogue, that the obstacles originating in this case, have rather a computing nature than a logical one, and hence they can be overcome due to utilization of the modern computers.

The system selection tasks, considered above have been constructed in a proposal, that the decision on the selection of "best" system, its parameters and control algorithm is realized once at the research design stage. The use adaptive conception permits to form the criteria of systems comparison, changing in the course of system operation on the basis of operation information, received during the flight.

Such the systems comparison criteria and the appropriate selection tasks only have initiated to be developed and the conduction a range of investigations is necessary for its discussion.

The simulation example of spacecraft intellectualized control system

Let us consider the landing control system of a spacecraft (Figure 4).

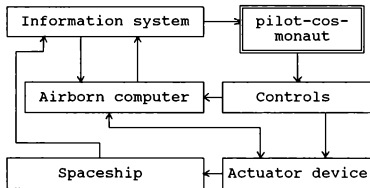


Fig.4. The landing control system of a spacecraft.

We shall model the system in which a pilot-cosmonaut realized the control on pitch channel on the basis of information, received from the information system (including visual aids).

The one of seven crew members is assumed to participate in the control (01, 02, ..., 07). Considering, that pilot control efficiency are characterized the control system characteristics - landing quality demands relation, that two control modifications for the comparison of two systems (S1, S2) are considered. The first modification of control assumes the availability of adaptive readjustment of the control parameters K_a for each crewmember, what complicates the system, as

required a storage in the computer memory of necessary collection of parameters for each crew member (K_a, \dots, K_a) and its input to the system for each operator, i.e. the first modification assumes the availability of intellectualized algorithms A1.

The second modification is based on the rigid selection of equal parameters for all operators.

The landing system simulation is realized at the hybrid complex (Figure 5).

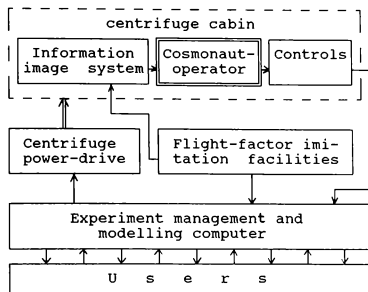


Fig.5 The scheme of the hybrid complex of spacecraft landing simulation.

A presence of the centrifuge enables under simulation process to create the g-load influence to a man-operator during the different descent paths and landing control. For more adequate g-load influence to the man-operator the special procedures, reproducing the human state in weightlessness before the start of simulation. The preliminary training of operators before the simulation rises a human perception of situation and control process under the spacecraft landing.

During the spacecraft landing the task of provision of high control accuracy and safety is developed before a cosmonaut.

Hence the probability, that system errors, an average rate of heart beats, breathing rhythm and blood pressure of the pilot-cosmonaut are failed inside permissible meaning, takes as the efficiency index of landing operation.

Two control systems (S1, S2) are simulated. The difference of systems consists in different methods of information presentation to the pilot-cosmonaut. In the first system S1 the pilot realizes a control on the basis of information about spacecraft attitude and final result prognosis, but in the second system S2 the pilot obtains an information only on an attitude and speed characteristics of the spacecraft motion. Quantitative meanings of the probability P for seven operators, two systems under use

of the adaptation algorithm and its absence (parameters were defined in S1 for O6, a in S2 for O4) are seen in Table 3.

Table 3.

Systems	Presence of adaptation	OPERATORS						
		O1	O2	O3	O4	O5	O6	O7
S1	yes	0.82	0.64	0.72	0.69	0.7	0.64	0.77
S2	yes	0.77	0.56	0.72	0.43	0.59	0.54	0.79
S1	no	0.76	0.6	0.72	0.62	0.69	0.64	0.76
S2	no	0.77	0.54	0.71	0.43	0.58	0.64	0.69

The analysis of results shows, that if the first operator at the first system has realized the control, that the "best" control is provided and $P(O1, S1, A1)=0.82$. However the guarantee meaning of the index for the first system is determined by the control by second and six operators and then $P(O2, S1, A1)=P(O6, S1, A1)=0.64$ i.e. in case of refusion of the "best" operator for the first system the efficiency decay consists 22%. For the second system the "best" control is realized by seventh operator and $P(O7, S2, A1)=0.79$, but under control by any operator the guarantee meaning of index $P(O4, S2, A1)=0.43$, i.e. the efficiency decay by 46%. Therefore the first system S1 is more effective than the second S2. It gives more stable meaning of the control efficiency under an operation of any crew member. At the same time the simulation result is interested, that third operator O3 provides for the same control at different systems, but seventh operator O7 has realized more effectively the control at second system, as $P(O7, S1, A1)<P(O7, S2, A1)$.

The analizable results of simulation show, that an introduction of adaptation for seventh operator O7, when S2 should switch upon his operation, practically leads to a creation of new system S3, which dominated the systems S1 and S2 on the accepted efficiency index. However the system S3 will have the same "best" and guarantee meaning of efficiency meaning P.

It will be noted, that the performed simulation have confirmed quantitatively, that the use of adaptation algorithm, i.e. the realization of the most simple intellectualized control allows to rise the efficiency of control as the meanings of indices for first and second systems with algorithm of adaptation dominate over the meanings of this index for a case, when the adaptation algorithm is absent. At the same time for separator algorithms the introduction of intellectualization does not increase the efficiency of control, what is confirmed not only the performed simulation, but also the experience of development of different man-machine control systems.

Coming to an end the discussion of investigation results performed it can be noted, that the proposed method of intellectualized control systems simulation opens the capabilities of detailed analysis of this system class and selection on the basis of comparison of the "best" system indices. The significant feature of proposed approach to system simulation and use of the sum total of indices consists in that, what the capability of parallel work with the modelling results in the dialogue regime of different specialists makes available for improvement of system quality, its used efficiency, as well as perfection of pilots training and learning.

References

1. The Spacecraft Semi-Automatic Control Systems simulation /G.T.Beregovoy, A.I.Yakovlev, V.M.Vasilets. - M: M., 1989. pp 278.
2. The efficiency of technical systems /Edited by Utkin V.F., Kryuchkov V.V., authors: Ohotnikov G.N., Torbin V.U., Yakovlev A.I. et al)
3. Yakovlev A.I., Gerasimov N.A. /Adaptive dialogue systems. M.: MO, 1986, pp 380.

WHAT OPTICAL CUES DO PILOTS USE TO INITIATE THE LANDING FLARE? RESULTS OF A PILOTED SIMULATOR EXPERIMENT

S.K. Advani*
J.C. van der Vaart**
R. Th. Rysdyk†
J. Grosz‡

Delft University of Technology
Faculty of Aerospace Engineering
Kuyverweg 1, 2629 HS
Delft, The Netherlands

Summary

A piloted moving-base simulator study of the landing of a twin-engined executive jet airplane was conducted to find out if pilots also use an optical variable called the Time-To-Contact (TTC) or tau, to time their actions. By manipulating the approach-path angle and the visual speed of the visible runway outline, the influence of the perceived TTC on the initiation of the landing was assessed. Results suggest that pilots indeed use some kind of Tau-margin strategy, but rely on the judgement of absolute height as well. Further experiments are needed to rule out any influence of prior training on the timing of the flare. Recent work on timing and perception suggests that the amplitude or speed of control actions may be determined by a higher order variable, i.e. the perceived rate of change of the Time-To-Contact, called Tau-dot. Possible implications of this for further work are mentioned.

Nomenclature

C_v	instantaneous vertical speed
f	visual velocity amplification factor
H_f	height of flare initiation
H_v	visually-perceived height above surface
h	magnitude of image projected on retina
x	distance to aiming point
V	instantaneous speed
V_f	sink rate at flare initiation
V_v	visually-perceived sink rate
δ_a	elevator deflection
γ	vertical angle to the aiming point
θ	aircraft pitch attitude w.r.t. ground
Ψ	optical angle of runway at focal point
τ	time to contact

* Assistant Professor, Member AIAA

** Assistant Professor

† Research Assistant, TU Delft

‡ Research Assistant, Vrije Universiteit Amsterdam

Introduction

The landing flare is a time-constrained maneuver. On the approach to land, a pilot establishes a constant sink rate of typically 2.5 to 3.5 m/sec (between 500 to 700 feet per minute) for jet transports and, after lining up the aircraft with the runway at a typical wheel height of about 7 to 15 m (20 to 50 ft), the landing flare is initiated by slowly pulling the control wheel backwards. Consequently, the phugoid motion of the aircraft is excited and the flare is essentially the first quarter part of the phugoid, in which height and flight speed are mutually exchanged. Hence, the sink rate is reduced and exchanged by a small decrease in flight speed. Since the phugoid period is typically about one half the flight speed (around 70 m/sec in the approach), and because it takes some time to bring the control wheel to the rearward deflection that is necessary at touchdown, the duration of the landing flare is in the order of 6 or 7 seconds.

The values given above roughly represent those for a jet transport. Naturally, the actual conditions vary according to the aircraft type and configuration.

Figure 1 shows some computer-generated time-histories of typical height and sink-rate in landing flares. The elevator deflection in this case is a ramp-type pull, which was found to be typical for landing maneuvers^{1,2}. The maximum decrease in sink rate of course depends on the maximum elevator deflection, but the instant in time at which the desired sink-rate is achieved should ideally be just when the wheels touch the runway.

It appears that pilots consider touchdown sink-rates of around .5 m/sec to be desirable. After the sink-rate has reached a minimum, see Fig. 1, it will start to increase again due to the periodic and weakly damped phugoid motion. Therefore, not only a flare initiated too late (too low), but also a flare that is started too early (too high) will result in a hard landing. If the flare maneuver has been overly strong (i.e. if the control wheel has been pulled too far backwards), the wheels may not touch the runway at all before the aircraft

starts to climb, meanwhile rapidly consuming available runway length. Finally, if a maneuver is initiated in time, but with insufficient control deflection, a hard landing will follow. The foregoing illustrates that a safe landing flare resulting in a firm but not too hard touch-down requires accurate timing and control by the pilot.

Timing the action:
How a pilot might 'see' the
right time to initiate the flare

The optic flow during landing³ is a key factor in establishing the initiation and magnitude of flare. The forward and slightly downward motion of an airplane approaching the runway prior to landing causes the visible flow field to expand globally, see Figure 2. The focus of expansion is where the momentary flight path vector intersects the runway. If the instantaneous distance to the runway along the flight path is $x(t)$, and the instantaneous speed along the flight path is $V(t)$ then, if the pilot would take no action, the aircraft would crash land after the time-interval:

$$\tau(t) = x(t) / V(t) \quad (1)$$

The variable τ is the instantaneous time to impact, or TTC, the "Time To Contact". Since τ is continuously decreasing with time, it is denoted as a function of time, $\tau(t)$. If the angle Ψ is, for instance, the optical angle specified by the runway width at the focus of expansion (see Figure 3), then it is easily shown - by approximation for small values of Ψ - that the following relation between the time-to-contact τ and the angle Ψ holds^{3,4}:

$$\tau(t) = - \Psi(t) / \dot{\Psi}(t) \quad (2)$$

where $\dot{\Psi}(t)$ is the rate at which the optical angle Ψ changes with time. Of course, the relation of Eqn. 1 holds for any angle Ψ subtended by an environmental structure in the visible field.

Many experiments have shown that tau information, as specified by Eqn. 2, is picked up or 'seen' directly by humans (and animals) and is used to guide or control their motions in the visible environment^{5,6,7,8}.

Hypothesis: pilots rely on TTC

It was hypothesised that, for visual landings, pilots use some kind of Tau-margin strategy to decide upon the initiation of the flare. In dynamic tasks such as the catching of falling balls, where the instantaneously observed τ does not specify the real time-to-contact (since falling balls accelerate), subjects gear their actions to the *observed* τ and not to the real time until collision⁹. This could imply that even in the case where the motion of the aircraft at the start of the flare

is non-stationary, pilots would also use tau information to gear their actions. The experiments described in this paper were designed to verify these notions which seem to be rather novel in the field of manned flight.

Experimental Setup

Simulation hardware

The TTC hypothesis was investigated in a real-time moving-base simulator experiment by evaluating the control actions of subjects in responses to manipulations in the visual stimuli. The three degrees-of-freedom (pitch, roll and heave) flight simulator of the Delft University of Technology, shown in Figure 4, was used in these trials. Note that during the flare/touchdown maneuver, the pitch rotation and the vertical translation heave are especially necessary. This simulator incorporates a transport aircraft cockpit with generic instruments, as shown in Figure 5.

Visual display system

Visual cues were presented by a night-only visual display system. A dot-pixel image generator driving a monitor-based collimation system provides a realistic representation of an ILS runway and surrounding lights (see Figure 5). Note that no VASI symbology was used; the aircraft was situated on the glide slope before commencing each run.

Flight Equations

The aircraft simulated in this experiment represented a twin-engine business jet airplane, namely a Cessna Citation - I. This particular model was selected since all available pilots were familiar with the characteristics of this type, and because the model has been accurately tuned throughout the flight envelope using parameter identification techniques⁹. Table 1 shows the important geometric properties of this aircraft. Control forces, presented through the hydraulic control loading system, were also identified for this type. Aerodynamic ground effect was omitted from this experiment to guarantee that the flare initiation was due to pilot input alone. This did not adversely affect the aircraft dynamics, and was not noticed by the pilots.

Simulator timing

The entire simulation ran at a thirty-Hertz update frequency. The visual display is based on analogue signal generation devices and hence introduces very little time delay in comparison with CGI techniques. Motion and control loading here are also analogue-controlled, thus having minimal time delays.

Experimental validation of the
time-to-contact theory during the
flare maneuver

Approach and landing simulation trials were conducted in various aircraft attitude and visual environment

configurations. The subjects were asked to recursively conduct an approach and landing using the Instrument Landing System: Glide slope and localizer information, coupled to the reference position of the outside image of the runway, was provided in the aircraft instrumentation. No flight director was used.

Two primary types of landing approaches were conducted: The first involved variations in the approach angle¹⁰, while in the second experiment the visual velocity of the outside world image was amplified or attenuated when the airplane descended below 50 metres altitude¹¹. Both of these served to determine the influence of the visual image flow field on the initiation of the flare maneuver.

Simulator trials began five kilometers before the runway threshold. The airplane was trimmed in a normal landing configuration at 110 knots IAS on the glide slope: 40° flaps, gear down, and throttle set for constant forward velocity. Pilots were asked to maintain the initial trimmed attitude and remain on the localizer/glide slope until they decided that the flare initiation should commence.

Recording of pilot control force inputs and aircraft variables as functions of time began one kilometer before the threshold, thus well before the flare.

Data logging

The measured data contained time histories of the variables x , h , vertical speed, τ , V , γ , θ , δ_e , and V_v . The moment of flare was determined by analyzing the elevator deflection signal δ_e .

A typical elevator control column deflection time history during flare is shown in Figure 6. All data was post-processed by Matlab programs.

Manipulation of approach angle

In the first experiment, the airplane was positioned on the approach path and configured in a trimmed condition, but placed initially at various heights. By following the glide slope indicator instrument and aiming for the normal touchdown point, the aircraft would fly on one of three approach angles: 2.0°, 3.0°, or 4.5°. The glide slope was limited to a maximum of 4.5° due to the aerodynamic properties of the Citation. At higher approach angles, the flight speed could no longer be held in equilibrium due to limitations in aerodynamic performance.

Since altitude divided by the vertical speed determines the time to contact with the ground, the consistency in the use by the pilots of the time to contact phenomenon could directly be determined from these evaluations.

Manipulation of "visual" airspeed

The determination of the use of visual cues during the flare was accentuated by amplifying or attenuating the forward

speed, hence the airspeed, of the simulated aircraft. If time-to-contact were the mechanism being controlled by the pilot (i.e. if the pilot would indeed time his actions to the perceived tau-margin), then the flare would be initiated by the perception of the projected net velocity vector (or airspeed): A higher airspeed and earlier flare, and a lower airspeed and a delayed flare would support this hypothesis. Therefore, the airspeed presented through the visual display system was increased, decreased, or was left unchanged.

In order to make this experimental manipulation unnoticeable to the subjects, the airspeed variation was only introduced as the aircraft passed below 50 metres altitude. Above this altitude, pilots were asked to fly (head down) on instruments alone. If the entire run had been manipulated in this way, the variations in the length of the trials would have been noticeable as this would effectively have represented a horizontal wind.

Seven airspeed manipulation configurations were possible, in which the airspeed (at the 50 metre "trigger" altitude) was presented as .75, .80, .90, 1.0, 1.10, 1.20, and 1.25 times the actual true airspeed. Note that only the outside-world visual speed reference was manipulated; instrument readings and motion cues were unaffected. Seven final trials were conducted in each configuration, resulting in forty-nine trials per subject.

Effect of moving visual runway on tau

If the tau-margin is determined by dividing the altitude by the vertical speed (sink rate), then the effective tau-margin in the case of a visually manipulated scene is given by

$$\tau_m = -H_v / (f V_{\text{visual}})$$

Determination of flare initiation

Although Delta-e appeared quite constant during the approach up to flare, it was necessary to discriminate the exact location of the initiation of the flare from other variations. A moving average filter (MAF) applied to the elevator control signal provided a reliable solution by doing the following: The variance of the entire signal is calculated. Then, an interval of the signal is analyzed, with the MAF "window" corresponding to the length of the interval. In this experiment a window length of five seconds was selected. The signal average is calculated over the window. The window is moved one step forward in time, and again the windowed average is calculated.

This process is repeated over the entire range of interest. When the deviations equal a multiple of the variance, the signal is considered to change. This event is marked, and the initiation of flare is thus determined.

Other methods, including linear and sinusoidal curve fits to the elevator deflection, and even visual inspection were used to verify the MAF technique.

The moment of touchdown was also recorded for the analysis. This was detected directly from the simulator software using the height above ground as the control variable.

Evaluation pilots

Three evaluation pilots participated in this experiment, all of which are commercial pilots with proficiency in the Cessna Citation I.

Training and learning

The subjects were initially trained to perform the landing in the simulator and, due to their high level of skill in this aircraft type, consistency was quickly achieved. Training ceased when both the subject and the experimenter felt that more training was not necessary, and when the glide path deviation from 1000 m to 100 m before the runway threshold was consistent. This deviation was also monitored during final runs to ensure consistency.

Discussion of Results

Locations of flare initiation

Figures 7 and 8 present the points associated with the initiation of the flare as determined by the MAF, described above. For clarity, the values shown represent one subject, but are indicative of the general trend. Furthermore, these values show that this subject indeed remained on the glide path until he decided that it was necessary to initiate the flare.

Effects of varied approach angle

Figure 9 shows the altitude versus sink rate for one of the three subjects when the approach angle was varied. A straight-line fit with the origin at zero would suggest a constant tau-margin. At high sink rates the data tends to be spread, however a regression analysis indeed suggests a tau-margin dependency.

The means and standard deviations of the flare altitude (H_f), the sink rate at the initiation of the flare (V_f) and the corresponding tau-margin are presented in Table 2. The vertical speed at touchdown is also given. A linear regression suggests a highly significant relation between the V_f and H_f , with $p < .001$. Although a clear relationship exists between H_f and V_f , H_f was not constant for different approach angles, supporting the tau dependency. However, the tau-margin appears to decrease with increasing approach angle.

It may be argued that the tau-margin is not exactly maintained in these evaluations, however H_f also varies with the approach angle. It appears that the pilots initiate the flare based on both the height above the runway and also on tau through a learned mental process. Note also that the sink rate at touchdown also increases with increasing approach angle, suggesting that the tau-margin dependency would be

stronger if special attention (by the pilots) were devoted to maintaining a consistent touchdown speed.

It would be valuable to manipulate the scene content (field of view, detail in foveal area, texture) in the image and evaluate the effect on flare action. Further comparisons of the data with actual flight test experiments would also be valuable in providing baseline figures. It has been suggested that sink rates at touchdown are higher in simulators than in actual aircraft¹². Motion cues resulting from the "touchdown bump" provide valuable sink rate information; a soft bump provides a false vertical speed cue. Simulator time delays would also influence the results and amplify the vertical speed at touchdown.

Effects of variations in visually-presented speed

In the second experimental scenario, the airspeed was amplified or attenuated by a factor to independently increase or decrease the relative motion of the outside image. This effect was triggered as the aircraft descended below 50 metres. A higher forward (visual) speed would result in an earlier flare initiation according to the tau-margin hypothesis. Pilot responses for various visually-perceived speeds are shown in Figure 10 and 11 for one representative subject.

The tau-dependency was not so pronounced at lower image speeds (Figure 10), although some relationship appears between altitude and sink rate as shown by the linear regression applied. The relation becomes much stronger at higher image speeds (Figure 11). These results suggest perhaps even a nonlinear relationship, a subject for future research.

Further continuation of research

The foregoing discussion clearly illustrates that the use of the tau-margin in flare is significant to pilot action. This research has provided a first insight into this phenomenon which is relatively new to the aeronautics field. Identification and quantification of tau and tau-dot in such actions provide an impetus for future research. Flight test measurements would provide a valuable datum, provided that the relevant parameters could be accurately measured. A subsequent simulator experiment could be conducted to try to match flight trials. Manipulation of the visual scene, either by static effects such as runway scene detail, or by dynamic effects like the manipulated image speed in this experiment, could then be further studied. This would serve to support the scene content requirements for simulators

The foregoing research focused on the use of an accurate aircraft model in a moving base simulator to deduce a particular perceptual process. Since this process is known to exist in other action perception events, it was the aim of this project to determine a correlation with the landing flare maneuver. It may however be valuable to simplify the experiment, for example:

- * Build a more fundamental dynamic model in which fewer dynamic variables play a role. For example, during the final segment of the approach it was observed that the pilots make fine adjustments to the longitudinal trim setting, which could have consequences on the elevator deflection applied during the flare. Furthermore, such a model would be capable of making very steep approaches, and could even operate over a very wide speed range.
- * Automatically maintain the airplane on the correct approach path and require only the initiation of flare by the pilot.

Conclusions

1. Time-to-contact plays a significant role in knowing when to initiate the flare. The tau-margin is however not the only factor in the perception process. From these experiments, both the height above the runway and the tau-margin determine when the flare is initiated. Both the varied approach angle and the manipulated visual image experiments supported the tau-margin hypothesis, and the notion that height also filters the usage of tau.
2. Determination of the moment of flare from recorded simulator data was quite consistent by using the moving average filter technique.
3. Since the flare maneuver is directly coupled to the vehicle dynamics, and also to its control system, the pilots quite certainly learn how much time-to-contact information should be used during a particular situation.
4. Pilots are trained to make consistent approaches, on a prescribed glide slope and at a reference speed. Large variations from these conditions may trigger inappropriate perception mechanisms, such as in the high approach angle cases studied here.
5. Knowing that time-to-contact is perceived through visual channels, and that this parameter is essential in timed event perception in general, then the quality and quality of the simulator visual scene play a role in the flare maneuver.
6. By establishing and identifying the perception process during the landing phase, it is possible to then determine the role which the elements of the visual scene play in ensuring fidelity between piloted flight and piloted flight simulation. Time-to-contact can provide a meaningful element by which performance can be measured. In such cases, it would also be advantageous to determine where the eye is looking (at the aiming point, or at the horizon) by means of eye-tracking equipment.
7. The time-to-contact element of the human perception

process can be further exploited in future aviation and also non-aviation displays. In particular, displays which present predictive information could benefit from knowledge of internal human perception models in order that external cues do not conflict with these internal mechanisms.

Acknowledgements

This was the first bilateral research project conducted jointly by the Delft University of Technology Faculty of Aerospace Engineering and the Vrije Universiteit (VU) Amsterdam Department of Human Movement Sciences. The authors would like to express their thanks to Dr. R. Bootsma, Dr. P. van Wieringen, both from the VU, who acted as advisors. It is the intention that this cooperation will accelerate in view of new initiatives in Delft, namely the International Centre for Research in Simulation, Motion and Navigation Technologies SIMONA, and the recent purchase of a Cessna Citation II research aircraft (jointly with the National Aerospace Laboratory NLR).

The authors would also like to thank B. Benard, R. Brons and J.J. van Hemel who participated as evaluation pilots, and Professor J.A. Mulder, ir. R. van Paassen and ing. H. Lindenburg for their valuable technical support.

References

1. J.C. van der Vaart, "The landing manoeuvre of a single-propeller aeroplane, a subsonic jet transport, and a supersonic transport, (in Dutch). Delft University of Technology, Department of Aerospace Engineering, Report VTH-144, 1967.
2. C.H. Reede, "KLM-studies on the landing of aeroplanes in adverse weather conditions, (in Dutch). De Ingenieur, 77, no.11, L1-14, 1-13, 1965.
3. Lee, D.N., "Visual information during locomotion". In: R.B. McLeod, H. Pick (Ed's), "Perception: Essays in honor of J.J. Gibson", 250-267. Cornell University Press, 1974.
4. Lee, D.N., "A theory of visual control of braking based on information about time-to-collision". Perception, 5, 481-501, 1976.
5. Bootsma, R.J. and Wieringen, P.C.W. Van, "Timing and attacking forehand drive in table tennis". Journal of Experimental Psychology: Human Perception and Performance, 16, 21-29, 1990.
6. Lee, D.N., "Visio motor coordination in space-time". In: G.Stelmach, J. Requin, (Ed's), "Tutorials in motor behavior", 281-293. Amsterdam, 1980.

7. Horst, A.R.A. Van der, Brown, G.R., "Time-to-collision and driver decision making in braking". Report IZF C-23, TNO Institute for Perception, 1989.
8. Lee, D.N., et. al., "Visual timing in hitting an accelerating ball". In: Quarterly journal of experimental psychology, 35A, 333-346, 1983.
9. Baarspul, M. and Mulder, J.A., "The composition of flight simulation models: Flight testing versus DATCOM techniques". AIAA-90-3120-CP. (From AIAA Flight Simulation Technologies conference proceedings, 1990).
10. Grosz, J., "Flying, the most rapid kind of locomotion", Vrije Universiteit Amsterdam, Department of Human Movement Sciences, M.Sc. Thesis, September 1991.
11. Rysdyk, R. Th., "Visual perception in a simulated landing task", Delft University of Technology Faculty of Aerospace Engineering, M.Sc. Thesis, August 1991.
12. Randle, R.J. et al, "The use of total simulator training in transitioning aircraft carrier pilots: A field evaluation". NASA TM-81250, 1981.

Mass	4536 kg
Approach speed (V_{approach})	51.4 m/s
Flap setting	40°
Mean Aerodynamic Chord (M.A.C.)	2.022 m
Wing area (S)	24.2 m ²
Centre-of-gravity location	0.30 M.A.C.
Aspect Ratio	7.38
Wing span	13.36 m

Table 1.Characteristics of Cessna Citation-1 in simulator landing trials

γ (deg)		H_i (m)	V_i (m/s)	τ (s)	sink rate at touchdown
2	Mean	10.27	1.63	6.21	0.84
	S.D.	3.07	0.52	1.75	0.26
3	Mean	13.09	2.31	5.70	0.86
	S.D.	3.93	0.31	1.53	0.52
4.5	Mean	17.18	3.82	4.52	0.99
	S.D.	5.35	0.71	1.27	0.57

Table 2. Statistical results for variations in approach angle

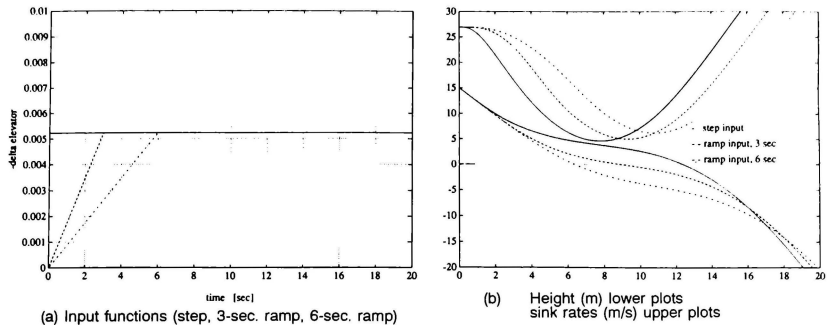


Figure 1. Typical height and sink rates during the landing flare of a jet-transport

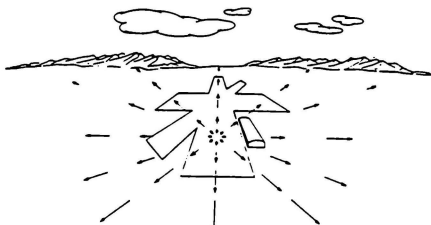


Figure 2. The outflow of optic array in a landing approach

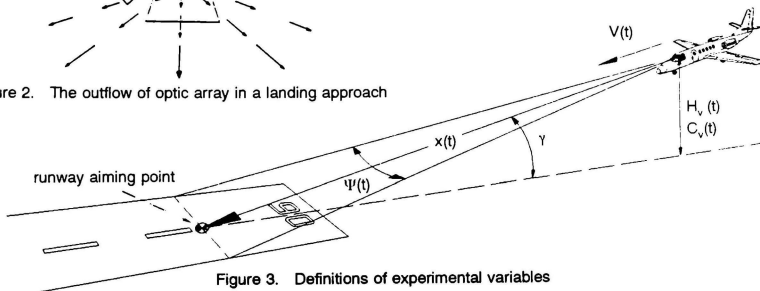


Figure 3. Definitions of experimental variables

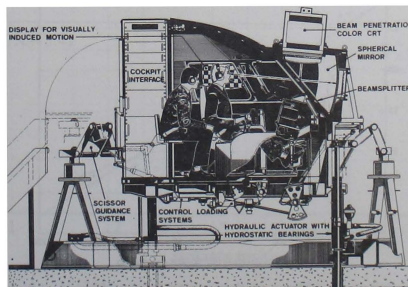


Figure 4. Cross-section of three-degrees-of-freedom flight research simulator

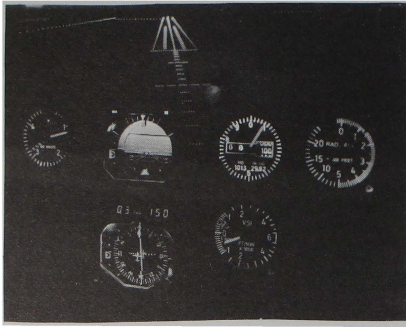


Figure 5. Instrument display and superimposed runway image during night landing

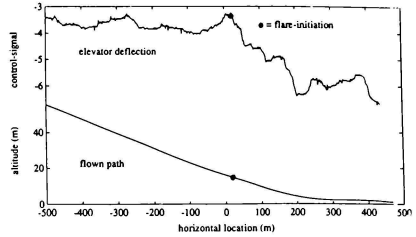


Figure 6. Measured elevator deflection time history during flare (typical)

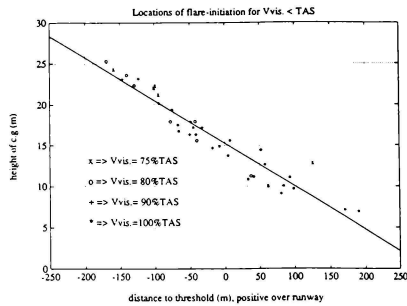


Figure 7. Initiations of flare as determined by the MAF: $V_{vis} < V_{actual}$

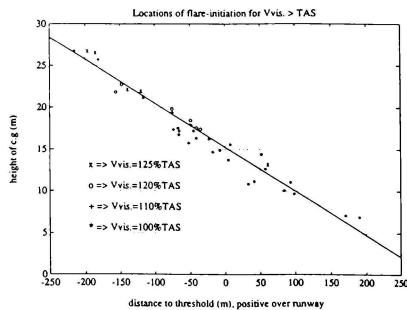


Figure 8. Initiations of flare as determined by the MAF: $V_{vis} > V_{actual}$

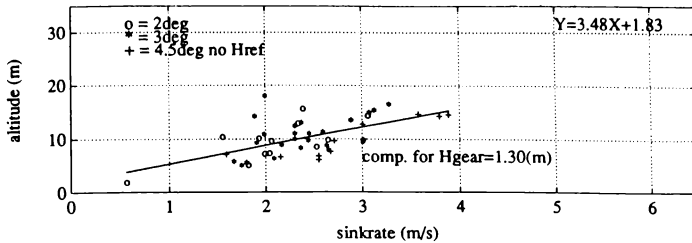


Figure 9. Altitude versus sink rate at flare initiation, approach angle varied, subject no. 1

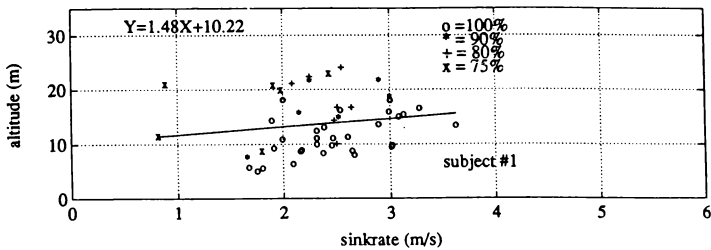


Figure 10. Altitude versus sink rate at flare initiation, variations in visually-perceived speeds: $V_{vis} < V_{actual}$

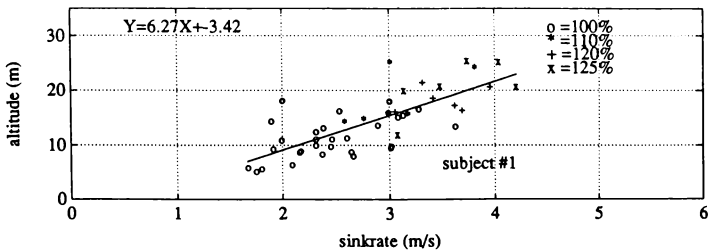


Figure 11. Altitude versus sink rate at flare initiation, variations in visually-perceived speeds: $V_{vis} > V_{actual}$

Acoustical and Vibratory Stimuli Interdependencies and their Applications in Simulation and Cue Synchronization

Robert J. Sawler* and Ron Matusof*
CAE—Link Corporation Binghamton, NY

Abstract

Over the years, the interest in cue synchronization and cue correlation has virtually ignored the measurement of acoustical and vibrational cues. The performance measurements of the simulator's secondary cues have many unique issues not previously addressed. This paper focuses on two of the secondary cues: the aural cue system, and the vibrational cue systems. This paper provides a brief look at classical cue synchronization and cue correlation of the primary cues. Next, it introduces the issues concerning the interdependencies between the aural cue system and the vibrational cue system. Finally, it discusses the importance of the secondary cues to the cue synchronization and cue correlation in simulation.

Introduction

The classical definition of dynamic response and cue synchronization has focused on the control stick input to the simulated system and the kinesthetic cue responses of the simulated system. The close correlation of cockpit instruments, motion system, and visual system are representative of the fidelity of a simulator's dynamic response and cue synchronization performance. This performance is generally specified in terms of the latency between control inputs and the primary cues. The primary cues are determined along the axes of interest for a given simulator, and may be readily measured from the motion, visual, and instrument dynamic responses. By measuring these dynamic responses, the interactions which a crew member can have with the simulated environment are evaluated. The correlation of the primary cues is extremely important to promote positive transfer of training and reduce simulator sickness.

The techniques for measuring the dynamic response and cue synchronization performance for single simulator systems has been well documented. These tests are often tedious, difficult, and time-consuming. The testing methodologies vary throughout the industry and include "pilot in the loop" stick inputs, high energy inputs into the aerodynamic models, and even automated cue synchronization testing.

At this time, there are no specifications which have been developed to include measurement of the secondary cues, such as aural and vibrational cues. The requirement for such a specification would depend on the level of fidelity of the training environment in which the simulator is to be operated. If the level of fidelity of the training device is low, such as a part task trainer, then the need for the correlation of aural cues and vibrational cues may be minimal. If the level of fidelity of the training device is high, such as a combat mission simulator, then the cue synchronization and cue correlation of the secondary cues becomes much more important. These requirements also become more involved when the scenario of simulator networking is evaluated.

Throughout the simulation industry, many tests are performed during the development and acceptance phases on the individual systems of the simulator. For example, the motion system and the visual system are tested for their individual performances, and then they are tested for their system cue synchronization. The secondary cues, however, are only tested to the individual system level. During the testing phase of a recent simulation program, a very interesting correlation was found between the aural and vibrational cues. The degree to which aural cues were provided appeared to have a significant effect on the amount of vibrational cues that were required. Although our study of this phenomenon was unscientific and relies solely on anecdotal information, there was a clear relationship evidenced between these two cues.

Auditory Cues

We rely on auditory cues for a variety of mundane tasks, such as telling us when to shift gears on a standard transmission automobile or when to remove our half-eaten left-overs from the microwave oven. We are more aware of information received from our proprioceptive and visual senses than we are of information received from our other senses, and we relegate the auditory sense to a status of a "secondary" cue. Though not generally considered as "primary" cues, auditory cues nonetheless provide important information which is unavailable via other senses.

* AIAA Member

It is generally accepted that auditory cues increase the transfer of training from the simulation to the real-world, yet there is little research to support this supposition. Research concerning state-of-the-art audio theory, especially in the area of psycho-acoustics, is surprisingly rare when compared with research concerning visual perception theory. It is interesting to note, for example, that there have been relatively few studies concerning the issues of auditory localization but there has been significant research directed toward visual localization.

A major concern of the simulation designer is to reduce, or ideally eliminate, apparent motion. Apparent, or illusory, motion is a general sensory phenomenon because it occurs in the visual, auditory, and tactual modalities.¹ However, the great emphasis in the study of illusory motion has been related to visual perception, and auditory apparent motion has received only minimal attention. Even when it has been addressed, auditory illusion has usually been discussed only in terms of an accompanying visual illusion.

Similarly, studies concerning cue correlation and cue synchronization have tended to be limited to the primary kinesthetic and visual cues of a simulation. Cue synchronization is defined as a measurement of the difference between the dynamic response of each tested system on a simulator. Cue correlation is defined as the degree to which the temporal sequencing of cues in the simulator matches the sequencing of the same cues in the real world. It is a widely held belief that when perceptual cues are in conflict, either within the simulation (cue synchronization problem) or between the simulation and the real-world (cue correlation problem), simulation sickness results. Simulation sickness symptoms are believed to occur when the sensory feedback associated with voluntary motion does not match expectation because of a sensorimotor rearrangement (such as prism glasses which reverse or invert objects).² However, this hypothesis is usually stated relative to visual and kinesthetic cues.

Auditory cues provide apparent motion cues as well. In the design of "space theaters", acoustic engineers are often concerned with producing a sound image which corresponds completely to the visual image. Five factors control the perception of sound: spectral information, loudness, direction, size, and distance.³ Spectral information and loudness contribute to how closely a simulated auditory cue correlates to its real-world counterpart (that is, the "reality" of the sound), while direction, size, and distance contribute to how that sound is spatially perceived. Research into surround-sound design principles has shown that the perceived

direction of an audio signal is strongly related to the temporal sequencing of simulated reflections. When a simulated reflection emanating from an independent set of speakers, lags the direct sound by as little as 12 milliseconds⁴, spatial perception is altered. Failure to synchronize these effects with the visual and kinesthetic cueing systems, or failure to correlate these cues with the real world can be quite disturbing and distracting to the simulator user.

Simulator designers are often dumb-founded to hear from their customer that the vibration or motion envelope of the simulator is not the same as the aircraft even though the designer has faithfully replicated real-world data in his simulation. Even if an individual cue, such as the vibration in a seat shaker, perfectly correlates with the real-world, an experienced crew member may not perceive the seat shaker as being properly modeled. A leading hypothesis of sensorimotor rearrangement states that the human spatial orientation system is normally calibrated to known conditions. For example, we know that lightning and thunder always come from the same direction, and we know the relationship between the lightning flash, the delay before the thunder is heard, and the loudness of the thunder. We therefore predict the occurrence of the thunder before it occurs. When known conditions are violated, sensorimotor rearrangement occurs. When this happens, the sensorimotor systems are disrupted and often produce erroneous results. In the simulator, one sensorimotor system may be perfectly stimulated (as is the case with the seat shaker designed by our dumb-founded engineer) while another is severely deficient in representing the real world (perhaps the aural cueing system). The result quite possibly could be perceived as an error in the perfectly reproduced sensorimotor cue and no perceived deficiency in the other.

Vibration

Vibration has also been relegated to the status of a "secondary cue", but it too provides important information for the sensorimotor interactions. Unlike the auditory cues, numerous studies have been conducted in the areas of vibratory and motion cues. When viewed as a whole, these studies are somewhat inconclusive as to whether vibrational simulation is necessary or provides a positive transfer to the real-world. Upon close examination, however, it would appear that many of these studies are flawed since they did not take into account sensorimotor rearrangements occurring due to visual system limitations, lack of aural cues, and the like.

A vibratory cueing system is a system whose purpose is to provide illusory motions within the simulation environment which are representative of the real-world vibra-

tions. As such, they are not necessarily replications of the real world but provide appropriate feedback such that an operator perceives the vibrations to be identical to the real world. Recent studies by noted authors in the field of motion and vibratory cueing have indicated that vibrational cueing, at a minimum, is necessary in cases of disturbance motion simulations (such as atmospheric disturbances).⁵

Vibratory stimuli are made up of both a frequency component and an amplitude component, and can be perceived both through proprioceptive and auditory senses. Vibration, as perceived through the proprioceptive sense, provides little cueing information toward the directional source of the vibration. Although both frequency and amplitude are important parts of vibratory cueing, the human's perception of the vibration's reality is more sensitive to changes in frequency. This would tend to suggest that important vibratory information might be reinforced or destroyed based upon its interaction with the pressure waves generated by the simulator's auditory cueing system.

Interactions between Vibratory and Auditory Cues

In many ways, vibratory and auditory cues are unlike other sensory cues. These cues are highly dependent upon the medium through which they travel. For example, the speed that an audio wave propagates through clear air is slower than the same wave's propagation through a more dense transport medium. Since directionality of audio is influenced by the temporal sequencing of the waves propagated from each source, the material characteristics of the simulator, and the material's relative location to the observer, influence the perception of the audio. Similarly, the material's characteristics also have a resonant frequency which, if properly vibrated, cause emanations of audio. It can therefore have an effect on the perceived directionality and type of auditory cues.

The existence of an interdependency between auditory and vibrational cues should come as no surprise. An acoustical wave is simply a vibratory wave transported through a less dense medium (and sensed through a different mechanism). Since this is the case, audio waves contain the two basic components of a vibratory cueing system: frequency and amplitude. Vibratory waves contain the five basic components of information from which we can derive an audio cue: spectral information, loudness, size, distance, and direction. So it is natural to talk about the interdependencies of these two cues, if we reference the effects of vibration on the ear and the effects of audio on the proprioceptive sensors of the haptic and vestibular systems.

Vibratory cues by themselves are not strong indicators of direction. In a series of experiments conducted at Miami University in Ohio⁶, it was found that the introduction of audio transients caused a perceived visual field shift. Introducing head vibration tended to attenuate the magnitude of the field shift, and the degree of attenuation was equal for both the right ear and left ear, and for both homolateral motion (motion toward the stimulated ear) and contralateral motion (motion away from the stimulated ear). The conclusion was that vibration elicits a middle ear reflex, which is bilateral in nature (i.e., there is poor discrimination of the vibratory source direction). We would therefore assume that in a simulator, direction of vibratory motion is sensed mostly from auditory cues. This is substantiated by several research programs which have found that illusory motions could be demonstrated under monaural listening conditions, but that only binaural listeners could distinguish the direction of the illusory motion.⁷

The apparent motion caused by the vibratory cueing system can also affect the perceived spectral information of the auditory system. In particular, simulated Doppler shifts, which are used to create the illusion of auditory distance, are modified by an actual Doppler shift caused by the vibratory cue. At higher frequencies, this additional Doppler shift may be significant enough to alter the ability to lateralize the audio signal.

Audio cues by themselves are excellent indicators of vibration frequency and amplitude. When the motion picture "Earthquake" was released, the concept of "sense-around" was said to be a major breakthrough in motion picture processing. In reality, "Earthquake" was simply the first careful design of theater audio to stimulate the vestibular system. The motion picture soundtrack used carefully chosen frequencies and amplitudes to create sound pressures within the theater that were spatially similar to those of an actual earthquake. Audience members "felt" the vibrations, though most of the vibration was illusory.

In combination, auditory and vibratory stimuli can either be additive to create a more realistic simulation or they can be destructive. The perception of the direction to an auditory source is due to the stereophonic nature of the human body (we have two ears) and the shape of the earshell. The ability to discriminate the audio source location is highly dependent upon head position. Therefore, a vibratory system which moves the head in the horizontal plane could seriously alter the perception of auditory direction unless the vibratory and audio cues were properly correlated with the real world.

In an informal study of crew perceptions of simulated cues, we discovered that the separation between vibra-

tional and aural cues was indistinguishable by the crew members. In particular, deficiencies in aural cues (such as when the instructor selected an audio level which was too low) were interpreted by the crew members as insufficient vibrational cues. When the audio intensity was increased, the vibrational cues were determined to be significantly more realistic. During a customer acceptance, the tendency is to tailor the cues to match the pilot's expectations of his perceptions. By not carefully considering the interdependencies between the various sensorimotor cues (i.e., vibrational and auditory cues, in addition to motion and visual), the tailoring may be erroneous and may even cause negative training or simulator sickness.

Discussion

The primary cues (kinesthetic, visual, and instruments) have often been regarded as the important criteria in determining total system dynamic response, system latency performance, and cue correlation. Whether or not this is a fair judgement, it is these cues to which a simulator's fidelity is measured. Often, a simulator which has good performance specifications in terms of the primary cues is also guilty of having high incidents of simulator sickness.⁸ The apparent inconsistency between good system performance (in terms of latency, etc.) with high incidence of simulation sickness may be resolved by examining the secondary cues. In particular, the interrelationship and interdependencies between the secondary cues has been shown to cause sensorimotor rearrangement. There is also strong evidence that the stimulation of the vestibular and visual senses have a strong impact upon the processing of secondary cues.⁹ It is therefore important to continue research in the relationship between secondary cue correlation and simulator sickness.

Recent discussions in the world of advancing technologies have centered on teleoperation and virtual environments (commonly referred to as "Virtual Reality"). Virtual environments highlight the need for correlation of both primary and secondary cues. Virtual environments are immersive — that is, they occupy all of the human's senses and often place the operator in a world where spatial orientation does not map to the real world (that is, the physical constraints of the real world do not necessarily apply). In these virtual environments, the correlation of all cues which affect sensorimotor operations

is critical. Although virtual environments conjure an image of a very futuristic technology, the problems of immersive environments are real, and need solutions today. The teleoperator controlling an object in space may be operating in a frame of reference which is different in spatial orientation than the terrestrial world. Since the operator's spatial orientation system is mapped to terrestrial constraints, sensorimotor rearrangement may occur, causing space sickness or loss of performance capabilities until such time as the operator's system has adjusted to the violation of terrestrial constraints. Proper correlation of primary and secondary cues, or introduction of correlated synthetic cues into a real environment may be a means of reducing serious systematic problems in virtual environments.

One might view the inclusion of secondary cues as an unnecessary complication to an already complex cue correlation and cue synchronization issue. This is not necessarily the case. Cue correlation and synchronization are system design issues, which transcend a given set of disciplines. The goals of simulation can be defined in terms of accuracy and realism.¹⁰ Accuracy refers to the precision with which a simulation recreates the real world. Realism refers to how closely a simulation actually correlates to the real world in both appearance and behavior.¹¹ For training applications, it is realism which is important. A realistic simulation is one which gives the illusion of an accurate representation of a real-world system. Simulation design engineers must look at the system design implications of all potential sensory cues and determine their impact on the realism of the simulation and their applicability toward a particular training goal. They must do this while staying sensitive to the overall health of the trainee (such as exposure of the crew to whole body vibrations).

As we stated at the onset, there is currently no set of standards which govern the cue correlation and synchronization of secondary cues. Before any such standard can be developed, more research must be conducted to ascertain the effects of secondary cues on overall system correlation and synchronization.

The overall performance of a simulation is only as good as its weakest link. Any efforts to produce high fidelity cue correlation will be incomplete if the dynamics introduced by secondary cues are not properly addressed as well.

References

1. Strybel, T.Z., Witty, A.M., and Perrott, D.R., "Auditory Apparent Motion in the Free Field: The Effects of Stimulus Duration and Separation", *Perception & Psychophysics*, 52 (2), 139–143
2. DiZio, P., and Lackner, J.R., "Spatial Orientation, Adaptation, and Motion Sickness in Real and Virtual Environments", *Presence*, 1, 3, Summer 1992.
3. Heringa, P.H., Kok, B.H.M., Dekeyrel, Y., "The Acoustics and Sound System for Hemispherical Film Projection", 77th Convention of the Audio Engineering Society, Hamburg, Germany, March, 1985.
4. Badger, G., and Davis, C., "Surround Sound in the Eighties -- Design Principles for Surround Monitoring Environments", 74th Convention of the Audio Engineering Society, New York, NY, October, 1983.
5. Cardullo, F.M., "An Assessment of the Importance of Motion Cueing Based on the Relationship Between Simulated Aircraft Dynamics and Pilot Performance: A Review of the Literature", AIAA Flight Simulation Technologies Conference, New Orleans, LA, August, 1980.
6. Parker, D.E., Gulledge, W.L., Perez, W., Poston, R., "Sound-Evoked Visual Field Shifts: Interaction with Five Classes of Stimulation", Report AMRL-TR-79-81, Air Force Aerospace Medical Research Laboratory, Wright Patterson Air Force Base, Ohio, January, 1980.
7. Strybel, T.Z., Mangias, C.L., and Perrott, D.R., "Monaural Listening Conditions", *Perception & Psychophysics*, 45(4), 371–377.
8. Kennedy, R., Fowlkes, J., and Lilienthal, M., "What Needs Doing about Simulator Sickness", AIAA Flight Simulation Technologies Conference, New Orleans, LA, August, 1991.
9. Cullen, J.K., Collins, M.J., Dobie, T.G., Rappold, P.W., "The Effects of Perceived Motion on Sound-Source Lateralization", *Aviation, Space, and Environmental Medicine*, 63(6), 498–504.
10. Barnes, A.G., "The Compromise Between Accuracy and Realism in Flight Simulation", AIAA Flight Simulation Technologies Conference, New Orleans, LA, August, 1991.
11. Miller, D.M., "Foundations for Tactical Training: A Challenge To Industry", AIAA Flight Simulation Technologies Conference, Boston, MA, August, 1989.

**TRANSPORT DELAY COMPENSATION: AN INEXPENSIVE ALTERNATIVE
TO INCREASING IMAGE GENERATOR UPDATE RATE**

Frank M. Cardullo*
&
Gary George**
State University of New York
Binghamton, New York

Abstract

Several delay compensation methods are investigated which illustrate that the phase lag due to system delays can be compensated in many cases. The paper compares the behavior of three methods; a lead/lag formulation by Ricard and Harris¹⁰, a predictor algorithm devised by McFarland⁶ and a state predictor algorithm developed by Sobiski and Cardullo¹³. The performance of these three algorithms is illustrated in response to system dynamics incorporating simulated aircraft dynamics, a pilot model and various visual or other subsystem delays. The results indicate the two predictor schemes are both considerably superior to the lead/lag. The differences between the two predictor algorithms are not quite as significant. However, the McFarland approach has substantial phase lag and gain distortion slightly above the design bandpass. However, the McFarland compensator is considerably simpler to implement.

Introduction

Virtually all real systems manifest a delay between the application of an input and the appearance of an output. The delay can be quantified with the transform pair, Laplace or Fourier, of phase shift in the frequency domain and pure delay in the time domain. Both the

time domain and frequency domain manifestations are of significance. The manual control literature contains abundant references to the effects of control system delays (1, 5, 8). The inclusion of a visual delay has the same effect on the pilot as a delay in the control system, primarily because, in either case, the manifestation of pilot action is delayed. In fact, in a simulator, delays in any of the elements (control system, aircraft dynamics or display dynamics) will be indistinguishable to the pilot.

Numerous studies have been performed to quantify the effect of simulator delays on pilot performance (1, 2, 5, 8, 9). In experiments involving target tracking, it has been demonstrated that significant degradation in performance due to visual delays occurs. The amount of delay that subjects tolerated decreased with an increase in task difficulty, but, when relatively complete motion cues were included in the simulation, considerably larger time delays were required to degrade pilot performance. In experiments conducted at the Armstrong Laboratory (11) employing both fighter and transport type dynamics, it was found that display delay degraded pilot performance. However, delays of 100 ms or less produced negligible effects. They also investigated the effects of display delay on training and found that there were essentially no effects for delays of less than 200 ms.

Figure 1 illustrates a system involving a pilot an aircraft and variable delay, which is represented by a Equation 1 (13). The delay is the time from a control input to visual manifestation of motion.

*Associate Professor, Department of Mechanical Engineering and Associate Fellow AIAA.

**Graduate Student, Department of Mechanical Engineering and Student Member AIAA.

$$\frac{\Theta(s)}{\delta_a(s)} = \frac{314.04(s^2 + 0.8929s + 3.46)}{s^4 + 7.1513s^3 + 9.163s^2 + 22.0625s} \quad (1)$$

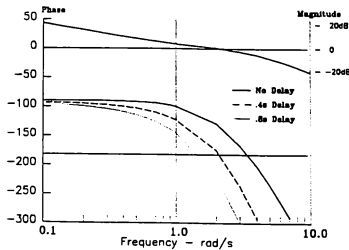


Fig. 1 Frequency domain effects of delays.

The figure shows the frequency domain effect of delays of 0, 400 and 800 ms. Notice that with no delay, the phase margin for the 2 rad/sec system is approximately 50 deg while at 400 ms it is almost zero and at 800 ms it is approximately -40 deg and, hence, unstable. Compensation, which will be discussed later, can recover phase margin lost to delay thereby improving handling qualities.

A viable simulation requires that not only the throughput delay but also the differences in delays among the various cuing channels be small enough not to degrade the man-machine system performance. Both result in poor pilot control and additionally may give rise to bizarre pilot perceptions, perhaps leading to simulator sickness. The contribution of visual delays to simulator sickness is less well documented than the effects on performance and transfer of training. To date, there appears to be only two studies which directly addresses this issue (4, 13).

Four basic mechanisms introduce delay as data passes through a simulation system: data sampling, data transfer, data processing and hardware latency. Sampling delay arises from the fact that in a digital simulation, time is

broken up into discrete frames during which the simulation inputs are sampled, the data are processed, and the resulting outputs are delivered to the cuing systems.

Data transfer delay is the time required to physically move data from one subsystem to another. The transfer time is significant only to the extent that it is time during which data is unavailable for processing. As long as the transfer time is less than the sampling interval (i.e., the frame length), transfer time may be considered the same as processing time. However, the data transfer itself may be conducted by a direct memory access (DMA) device separate from the host CPU so that if the timing of the transfers is appropriately chosen, the transfer time will not reduce the amount of time available for data processing. The processing time is the time required for the host computer to use any input data to update the aircraft state.

Discussion

To solve this problem, it may not be necessary to reduce the throughput of the image generator, which may be quite expensive. The solution begins with the systems engineering design of the simulator. Care must be taken to ensure that there are no additional latencies due to mistakes in program sequencing, unnecessary asynchronies between processors caused by incompatible update rates, or improper selection of integration algorithms. Once the system design has been optimized and there still exists an unacceptable delay in the visual system or other cuing system, then there are several techniques which can be employed to provide compensation for the delay. One approach is to use a lead/lag compensator, that is, a filter whose time constants can be chosen such that phase lag is compensated (10).

$$G_{LL}(s) = \frac{s + \omega_1}{s + \omega_2} \quad (2)$$

When employing this type filter, with the pole and zero specified by (10), the magnitude and phase relationships of Figure 2

resulted. This plot clearly shows the high frequency gain attendant to this formulation. This gain distortion is in the region of pilot control dynamics and, hence, is unsatisfactory. Notice also that none of the filters yield positive phase margin. Therefore, these filters result in an unstable system. Other attempts at implementing lead compensation of this type have been made, where phase margin was preserved, but the system still suffered gain distortion (3).

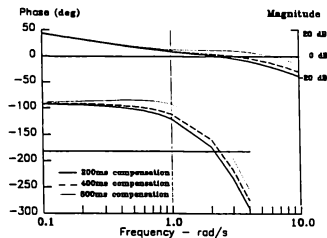


Fig. 2 Frequency response of system compensated by a lead/lag filter.

Another approach used to compensate for visual system delays is the approximation of a lead by a power series expansion. This method is discussed by several authors in the literature (7, 10). However, according to Dr. Grant McMillan, of the USAF Armstrong Laboratory, the major problem with this approach is that it appears to have noise amplification properties. A variation of the power series lead is the so-called "single interval lead" which employs the predictor capability of certain numerical integration algorithms to predict simulated aircraft position one frame ahead. A drawback of this approach is that it provides a maximum of one frame of lead.

A separate class of compensation is the

predictor algorithm of which two are discussed here. The first was developed by Richard McFarland⁸ at NASA Ames. According to McFarland⁸, this method produces negligible gain distortion while also minimizing phase lag. This algorithm is based on an extension of the sinusoidal integration algorithm which was developed at McDonnell Douglas Aircraft by David Rolston¹² of the McDonnell Aircraft. This algorithm is designed to capitalize on the oscillatory behavior of an aircraft in flight.

The McFarland compensation technique has been used on research and training simulators. The basic premise is the use of a velocity predictor which updates a current value of position. It uses two past values of velocity plus a current position value. The algorithm is (6);

$$U_{n+1} = U_n + b_0 V + b_1 V_{n-1} + b_2 V_{n-2} \quad (3)$$

where U is position and V is rate. It assumes that high-gain tasks are limited to two to three Hertz. By using sinusoidal inputs to the compensation delay model and tuning it at the bandpass, the coefficients b_0 , b_1 and b_2 can be determined in terms of delay, update rate and bandpass. These coefficients are calculated once and are constant for the real-time simulation. Hence, this yields a computationally efficient algorithm.

Analysis of the compensation for an update rate of 60 Hz delay of 67ms and a bandpass of 3 Hz is shown in the frequency response of Figure 3. Also shown is the uncompensated system. For this case it can be seen that for the compensated case the phase stays essentially at zero and provides phase lead around the bandpass followed by phase lag after that. The phase improvement over the uncompensated case is apparent. For the gain, the two cases follow exactly until the bandpass where the compensated system breaks abruptly upward. This gain distortion and phase lag above the bandpass in theory does not affect performance since the system components cannot respond. However some jitter in the visual scene has been observed.

The coefficients b_0 , b_1 and b_2 will have

an influence on the overall response. Since these values depend on time delay, update rate

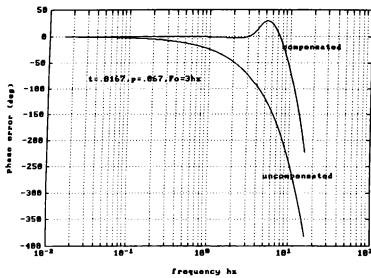


Fig. 3 McFarland⁶ Compensator with 67ms delay, 3 Hz bandpass and 60 Hz update rate.

and bandpass, Figures 4, 5 and 6 illustrate the responses for variations of these parameters.

As update rate increases (Figure 4), the amount of lead as well as the bandwidth of lead increases. Also, with lower update rates, the phase dips to lag before the bandpass increase. Gain for the presented values follow together with the uncompensated curve until the bandpass. With lower update rates, the gain turns back down at higher frequencies.

Figure 5 shows the effect of the bandpass value. Up to a certain point above 3 Hz, the amount and bandwidth of lead decreases with increasing band pass F_0 . For 5 Hz bandpass, the phase actually is negative and only recovers to zero phase in the bandpass area. Higher bandpass values cause further gain distortion at lower frequencies.

As can be seen in Figure 6, the increase in time delay causes more phase lag. Furthermore, as time delay increases, there is no phase lead at all (e.g. $P=.4$). Again, for some values, there is phase lag before the area of increasing phase lead at the bandpass. High delays cause considerable gain distortion as compared with lower delay values.

The McFarland compensator could be further improved over the full operating range

using an adaptive algorithm for the calculation of the rate coefficients. This might be done

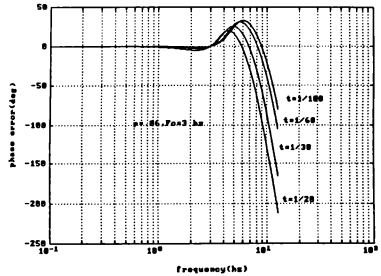


Fig. 4 The McFarland Compensator with various values of update rate.

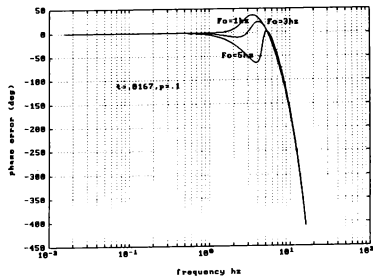


Fig. 5 The McFarland Compensator with various bandpass values.

based on the estimated state of the man-machine system.

A second predictor algorithm has been developed by Sobiski and Cardullo (1987)¹³. This predictive method uses the state transition matrix in a feedback loop to compensate for time delays, as shown in Figure 7.

Figure 8 illustrates the frequency response of the predictive filter discussed above. The magnitude plot retains the 2 rad/sec crossover frequency, and also the

integrator like crossover characteristic. In fact, close examination of the gain plot reveals that the -40 dB/decade knee of the lead filter and transfer function.

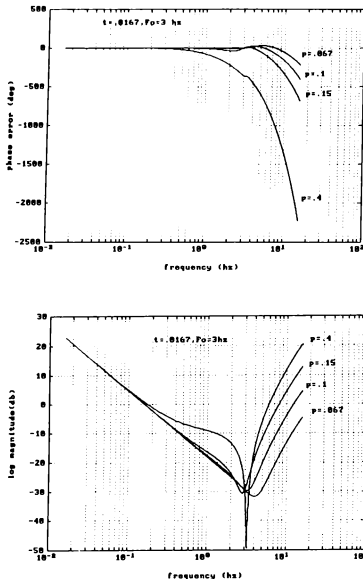


Fig. 6 The McFarland filter with various time delays (.067, 0.1, 0.15 and 0.4 sec.).

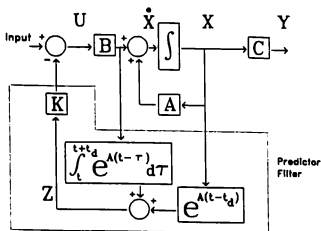


Fig. 7 State predictor filter block diagram.

the peaking of the McFarland filter do not exist. In addition, the phase shift of the zero delay case, illustrated in Figure 2, is essentially obtained for the three delay cases. The frequency response results indicate that the system damping ratio is not affected by the delays with the implementation of the state predictor algorithm.

Experiments were conducted by Sobiski and Cardullo¹² to determine the effects on performance of the state predictor. Figure 9 illustrates the effects on standard deviation of roll error for a roll axis disturbance nullification task. It can be seen that the full state predictor elicits essentially the same

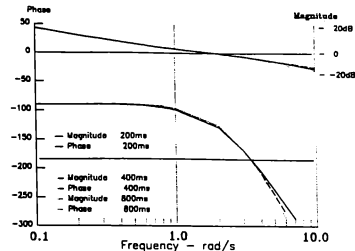


Fig. 8 Frequency response of system compensated by state predictor filter.

performance as the zero delay case while the lead/lag filter provides no discernible improvement over no compensation. In fact, the lead/lag at high frequency demonstrates worse performance than the uncompensated case at the longer delays.

In order to compare the performance of the McFarland compensator with other compensation techniques, such as those implemented by Ricard and Harris¹⁰ and the State Predictor by Sobiski and Cardullo¹³, it was necessary to test the compensation with a man-machine model. A transfer function for the variation in roll angle per deflection of the control stick was used for an aircraft model. The human transfer function was for a lateral control task performed with a rate controller. The combined transfer function is given in

Equation 1. Since the McFarland algorithm uses rate it was necessary to differentiate this

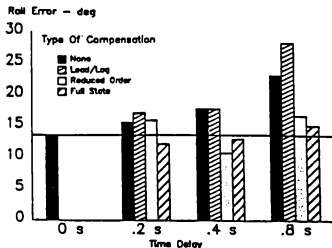


Fig. 9 Standard deviation of roll error in experiments employing the lead/lag filter and the state predictor filter.

Figure 10 illustrates the effect of the McFarland compensator on the full system (Equation 1) with a delay of 200 ms. Both the compensated and uncompensated systems are shown. Notice that the McFarland filter begins to introduce gain distortion slightly above two radians per second and a major amount of gain distortion above 10 rad/sec. These can be compared to the undelayed system shown in Figure 11. The McFarland compensated system as illustrated in Figure 10 shows zero phase shift below one radian per second and then maintains this ninety degrees of phase lead over the uncompensated system up to about 10 Hz. This result is somewhat troublesome since if the undelayed system of Figure 11 is considered, the system displays a 90° phase lag at low frequencies. Hence, the McFarland filter appears to introduce a phase distortion while compensating for the delay at low frequency. By contrast, the state predictor filter of Figure 8 appears to match the undelayed system frequency response quite closely while compensating the delay.

Conclusion

There is currently research being conducted to determine if providing motion

cues will compensate some of the visual delay. The conceptual basis for this research is that vestibular and somatosensory stimulation provide lead information, to the human operator, concerning vehicle state.

The McFarland compensator could be further improved over the full operating range using an adaptive algorithm for the calculation of the rate coefficients. This might be done based on the estimated state of the man-machine system. The results reported here indicate better performance of the state predictor algorithm over the McFarland Compensator. However, both perform substantially better than the lead/lag filter. A major advantage of the McFarland algorithm is its computational simplicity.

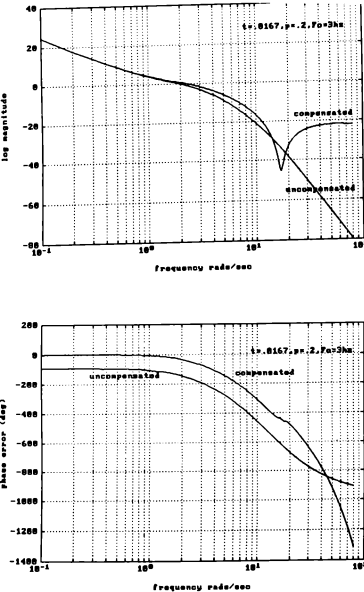


Fig. 10 The McFarland Compensator effect on the system dynamics of Equation 1 with an added 200 ms delay.

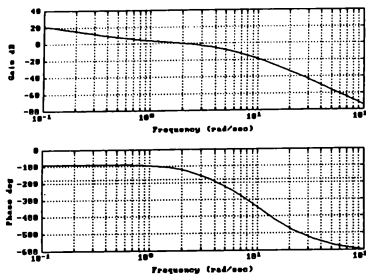


Fig. 11 The undelayed system dynamics of Equation 1.

The initial results are promising. It appears that proper simulator architecture, software design, compensation algorithms, and perhaps compensation through motion cuing provide a more cost effective approach to solving the visual delay problem than trying to further reduce the image generator throughput delay. If the effect of the delay can be removed, then the resources of the image generator could be employed to provide additional features or increase scene content.

References

1. Ashkenas, I.L. "Twenty Five Years of Handling Qualities Research." *J. Aircraft*, Vol. 21, No. 5, May 1984.
2. Cooper, F.R., Harris, W.T., and Sharkey, V. "Effects of Visual System Time Delay on Pilot Performance." *Proceedings 1975 NTEC Conference*, Orlando, FL, Nov. 1975, pp. 35-51.
3. Crane, D.F. "Compensation for Time Delay in Flight Simulation Display Systems." *AIAA Flight Simulation Technologies Conference*, Paper No. 83-1080, 1983.
4. Frank, L.H., Casali, J.G., and Weirville, W.W. "Effects of visual display and motion system delays on operator performance and uneasiness in a driving simulator." *Proceedings of the Human Factors Society Annual Meeting*, 1987.
5. Hess, R.A. "Effects of Time Delays on Systems Subject to Manual Control." *Guidance*, Vol. 7, No. 4, July-August 1984.
6. McFarland, R.E. "Transport Delay Compensation for Computer Generated Imagery Systems." *NASA JM-100084*, 1988.
7. McMillan, G.R. "Cue Integration and Synchronization." *Flight Simulation Update Lecture Notes*, State University of New York, Binghamton, NY, January 1993.
8. McRuer, D.T., Graham, D., Krendel, E. and Reisener, W. "Human Pilot Dynamics in Compensatory Systems." *Air Force Flight Dynamics Laboratory Technical Report No. AFFDL-TR-65-15*, July 1965.
9. Miller, G.K. and Riley, D.R. "The Effect of Visual-Motion Time Delays on Pilot Performance in a Simulated Pursuit Tracking Task." *NASA TN-D8364*, March 1977.
10. Ricard, G.L. and Harris, W.T. "Lead/Lag Dynamics to Compensate for Display Delays." *Journal of Aircraft*; 17, 212-7, 1980.
11. Riccio, G.E., Cress, J.D., and Johnson, W.V. "The Effects of Simulator Delays on the Acquisition of Flight Control Skills: Control of Heading and Altitude." *Proceedings of the Human Factors Society 31st Annual Meeting*, Santa Monica, CA: Human Factors Society, 1987.
12. Rolston, D.R. "Sinusoidal Integration for Simulation of Second Order Systems." *AIAA Flight Simulation Technologies Conference*, Niagara Falls, NY, June 1983, pp. 52-63.
13. Sobiski, D. J. and Cardullo, F. M. "Predictive Compensation of Visual System Time Delays." *Proceedings of the 1987 AIAA Flight Simulation Technologies Conference*, Paper No. 87-2434, Monterey, CA, August 1987.
14. Uliano, K.C., Lambert, E.Y., Kennedy,

R.S., and Sheppard, D.J. "The Effects of Asynchronous Visual Delays on Simulator Flight Performance and the Development of Simulator Sickness Symptomology." NAVTRASYLEN 85-D-0026-1, December 26, 1986.

MULTISTAGE INTEGRATION MODEL FOR HUMAN EGOMOTION PERCEPTION

Greg L. Zacharias* and Adam X. Miao
Charles River Analytics Inc.
Cambridge, MA

Rik Warren
Armstrong Laboratory
Wright-Patterson AFB, OH

Abstract

Human computational vision models which attempt to account for the dynamic perception of egomotion and relative depth typically assume a linear architecture for image processing. A common three-stage structure is: first, compute the optical *flow-field* based on the dynamically changing image; second, estimate the egomotion states based on the flow; and third, estimate the relative depth/shape, based on the egomotion states, and possibly a model of the viewed surface.

We propose an architecture more in line with recent work in human vision, employing *multi-stage integration*. Here the dynamic image is first processed to generate spatial and temporal image gradients which are then transformed into *snapshot* estimates to be used by the downstream egomotion state estimator. The estimator uses these snapshot estimates in combination with a *depth/shape estimate of the viewed surface*, to generate current state estimates. These, in turn, drive the depth/shape estimator, which employs an internal model of the world geometry, to generate an output which loops back to the state estimator. In this paper we describe this model, and its implications for modeling human egomotion perception, and compare model predictions with observed data.

1. Introduction

The problem of inferring egomotion and terrain shape from a rapid succession of two-dimensional motion images has been studied by perceptual psychologists and computer vision researchers using two distinctly different approaches.

The first approach studies problem empirically in some well defined and controlled contexts. For example, studies by Gibson^{1,2} Warren³, Grunwald and Kohn⁴, Eby⁵, Warren and Kurtz⁶, and Crowell and Banks⁷ concentrate on the effects of various environmental elements and human perceptual limitations on human egomotion estimation ability. Gibson^{1,2} showed that human visual systems have the remarkable ability to recover egomotion state from a moving image and established the *flow-field* explanation of human visual performance. It is postulated that, from the moving image an *optic flow-field* is generated, which produces a compelling sensation of egomotion and viewed surface shape. Experiments conducted by Warren³ indicate that the human can infer egomotion aim point with a 1.5 deg accuracy for straight and level flight above flat terrain. Grunwald and Kohn⁴ extended this experiment to curved motion and different textured terrain

samples. They also investigated the effects of velocity-to-height ratio, and obscuration of the visual field on egomotion estimation accuracy. Eby⁵ further found that viewed object shape estimation accuracy was: (1) jointly dependent on the speed at which an object moved and on the range through which the object moved; (2) not disrupted by nonsmooth optic flow-fields; and (3) more rapid for rotating objects than for curvilinearly translating objects.

All of these experiments were designed based on the flow-field hypothesis of the human visual process. Although these studies provide significant insight into the relationship between the human visual process and the environmental elements, their conclusions are highly context dependent and difficult to extrapolate to other situations. Moreover, the empirical approaches do not provide the predictive capability for situations where there does not already exist directly applicable data. Such a capability is clearly needed for effective design of novel simulator visual systems, for example.

The second approach seen in the literature formulates the egomotion and terrain shape estimation problem as a mathematical optimization problem, in which an observer is assumed to make egomotion and shape estimates by minimizing some well-defined loss function. Despite the apparent wide variety of approaches seen in the literature, almost all of the mathematical formulations use the common flow-field assumption, since it provides a convenient *divide and conquer* approach to studying human shape and egomotion estimation. One group of researchers can thus concentrate on flow-field estimation without considering its consequences for motion estimation, while another group can concentrate on the estimation of shape and motion state, based on the flow-field, without worrying about its generation.

There are a number of algorithms for computing the flow-field, given a sequence of dynamically changing images. The feature-based algorithm of Weng, Huang and Ahuja⁸ extracts the flow-field by matching features from one image frame to the next. The frequency-domain algorithms of Watson and Ahumada^{9,10}, and Heeger¹¹ work in the 3D spatio-temporal frequency domain (defined by the moving image's two spatial frequency axes and the one temporal frequency axis) to obtain a full bandwidth vector field defining the flow-field. Finally, the gradient-based algorithm of Horn and Schunck¹², Schunck^{13,14} uses an iterative method for solving differential equations to directly compute the flow-field. This algorithm first computes, at each pixel, the temporal and spatial gradients

*Member AIAA

of the image intensity function. A flow-field solution is then found which, in a least-squares sense, best satisfies a flow-constraint equation, while at the same time, maintains an artificially imposed *smoothness* over the sensor field-of-view. Further development is needed, however, to better account for occlusion boundaries and perceptual distortion, as noted by Schunck¹⁴. Recent efforts by De Micheli, Torre and Uras¹⁵ propose to use higher order gradients to compute the optimal flow to increase the accuracy of the flow-field estimate.

There are also a number of related algorithms to determine motion/geometry parameters, given the computed flow-field. The quasi-static estimator of Zacharias, Caglayan and Sinacori^{16,17} estimates an observer's instantaneous heading (vertical and lateral), angular velocity, and *impact times* (the speed-scaled range). Work by Broida and Chellappa¹⁸, Merhav and Bresler¹⁹, Sridhar and Phatak²⁰, and Sridhar, Cheng and Phatak²¹ demonstrate how modern dynamic estimation theory (in the form of Kalman filtering) can be brought to bear on the problem of generating *dynamic* state estimates. Young and Chellappa²² have shown how a 22-state extended Kalman filter (EKF) can be used to estimate the rotational and translational states of a moving object by a fixed observer, and it is clear that the dual of this estimation problem (moving observer in a fixed world) can be similarly formulated.

The artificial separation of flow-field computation and motion/shape estimation, however, has led to some unwanted consequences. First, the flow-field model can not predict the relationship between the true motion/shape states and the flow estimation process since the model isolates the *upstream* flow computation from *downstream* motion/shape estimates. Second, flow-field generation from *real* images (as contrasted with simplified feature-laden synthetic imagery) has been repeatedly shown to be subject to many sources of error and computationally intensive. Finally, the two stage approach simply requires too much computation.

This paper presents an alternative computational vision model, employing *multistage integration*. Here the dynamic image is first processed to generate the basic spatial and temporal image gradients that represent the image motion changes. This information is then directly sent to an egomotion state estimator and a depth/shape estimator. The state estimator uses the image gradient information in combination with a depth/shape estimate of the viewed surface to generate updated state estimates. These state estimates and the spatiotemporal image gradient information in turn drive the depth/shape estimator to generate updated depth/shape estimates, which in turn loop back to the state estimator. This feedback loop between state and shape estimators provides us with the desired multistage integration, and frees us from the artificial constraints imposed by conventional two-stage linear architectures. Our work has shown that this approach can significantly improve both estimation accuracy and computational efficiency, compared with the linear architecture approach. Moreover, model predictions are more in line with the recent empirical findings.

In this paper we compare predictions generated by our model with experimental results obtained by Warren³, Eby², and Grunwald and Kohn⁴. Without any artificially imposed sensor or processing limitations, our computational vision approach is shown to provide state/shape estimates which far out-perform the human observer, operating under comparable

cueing conditions. To more closely model less optimal human performance in egomotion estimation, we have taken a modeling approach which embeds some simple human perceptual limitations in the *front end* of our processing architecture. This then allows us to more closely match observed human performance, while reflecting human perceptual limitations in an integrated multistage structure.

2. Problem Formulation

Consider an observer undergoing both translational and rotational motion over a motionless terrain as illustrated in figure 2.1. The observer's linear and angular velocity are denoted by the vectors \mathbf{V} and Ω , respectively. Let P_i be a viewed point on the terrain, located at a range ρ_i from the observer. In the observer's frame of reference, the unit-length line-of-sight (LOS) direction vector $\mathbf{u}_i = \rho_i/\rho_i$ will appear to change with time, at a rate given by

$$\dot{\mathbf{u}}_i = \omega_i \times \mathbf{u}_i \quad (2.1a)$$

where ω_i is the rotation rate of the LOS vector, and is shown by Zacharias and Levinson²³ to be given as:

$$\omega_i = \mathbf{u}_i \times (\mathbf{u}_i \times \dot{\mathbf{u}}_i) - \frac{1}{\tau_i} (\mathbf{u}_i \times \mathbf{u}_{i-1}) \quad (2.1b)$$

where \mathbf{u}_v is the observer's *heading vector*, and τ_i is the *impact time*, defined by

$$\mathbf{u}_v = \frac{\mathbf{V}}{|\mathbf{V}|}, \quad \tau_i = \frac{\rho_i}{|\mathbf{V}|} \quad (2.2a,b)$$

The unit length heading vector defines only the direction of the observer's motion, but not his speed. The impact time is the elapsed time before the observer impacts with the surface at point P_i , if the observer were to head directly at P_i at the speed $|\mathbf{V}|$.

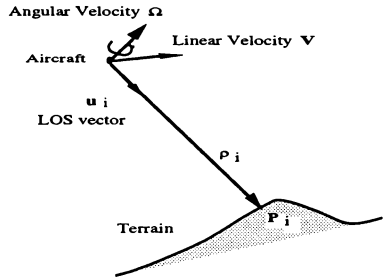


Figure 2.1: Line-of-Sight Rate

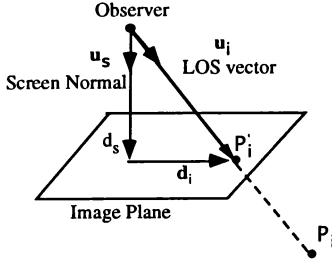


Figure 2.2: Image Plane Geometry

Consider now the that 2D image of the terrain formed on an image plane within the observer's field-of-view (FOV), illustrated in figure 2.2. Let the focal point be the origin of a screen coordinate system and let the effective focal length be d_s . The image of point P_i , given by P'_i , located by the vector d_i in the image plane, is defined by the focal-length normalized x-y coordinates $\delta_i = d_i/d_s = (\delta_{ix}, \delta_{iy})^T$. As shown by Zacharias, Miao and Riley²⁴, the rotation rate of the LOS vector u_i generates a rate of change of this in-plane location vector δ_i :

$$\begin{aligned} \dot{\delta}_i &= \frac{\omega_i \times u_i}{(u_i \cdot u_i)^2} \\ &= \begin{pmatrix} \delta_{ix}\delta_{iy} & -(\delta_{ix}^2 + 1) & \delta_{iy} \\ (\delta_{iy}^2 + 1) & -\delta_{ix}\delta_{iy} & -\delta_{ix} \end{pmatrix} \Omega + \begin{pmatrix} -1 & 0 & \delta_{ix} \\ 0 & -1 & \delta_{iy} \end{pmatrix} \frac{u_v}{r_{iz}} \end{aligned} \quad (2.3)$$

where u_s is the unit length normal defining the sensor plane.

Now, consider how the intensity I changes with time at an in-plane location δ_i . Using the chain rule, we obtain the following expression for the total temporal derivative of image intensity:

$$\frac{dI(\delta_{ix}, \delta_{iy}, t)}{dt} = I_{ix} \frac{d\delta_{ix}}{dt} + I_{iy} \frac{d\delta_{iy}}{dt} + I_{it} \quad (2.4)$$

where $I_{ix} = \partial I / \partial x$ and $I_{iy} = \partial I / \partial y$ and we have assumed the image brightness function is first order continuous at the image point δ_i . Assuming that brightness of any terrain point is constant over time, we require the total derivative of (2.4) to be zero, so that:

$$I_{ix} \frac{d\delta_{ix}}{dt} + I_{iy} \frac{d\delta_{iy}}{dt} + I_{it} = 0 \quad (2.5)$$

A more rigorous derivation of this basic flow constraint equation is given in Horn and Schunck¹².

Substituting (2.1) and (2.3) into (2.5), we obtain the following gradient constraint equation that relates the 2D image brightness I_i (through its derivatives I_{it} , I_{ix} , I_{iy}) with the 3D terrain shape information set $\{\tau_{iz}\}$ and observer motion states $\{\Omega, u_v\}$:

$$I_{it} = h_{it}^T \Omega + \frac{1}{r_{iz}} h_{iv}^T u_v \quad (2.6a)$$

where we introduce the image measurement vectors

$$h_{it} = [(-I_{ix}\delta_{ix}\delta_{iy} - I_{iy}(\delta_{iy}^2 + 1), I_{ix}(\delta_{ix}^2 + 1) + I_{iy}\delta_{ix}\delta_{iy}, -I_{ix}\delta_{iy} + I_{iy}\delta_{ix})]^T$$

$$h_{iv} = [I_{ix}, I_{iy}, -I_{ix}\delta_{ix} - I_{iy}\delta_{iy}]^T \quad (2.6c)$$

and where we introduce the image plane normal impact time (projected onto the z axis of the image plane):

$$\tau_{iz} = \frac{\tau_i}{\sqrt{\delta_{ix}^2 + \delta_{iy}^2 + 1}} \quad (2.6d)$$

The detailed derivation of these equations is reported separately in Miao and Zacharias²⁵. The above equations assume a moving observer and a rigid terrain. For the opposite situation where a stationary observer views a moving object, the same gradient equation will hold. Consequently, in the following any results obtained for the moving observer viewing a fixed world also hold for a fixed observer viewing a moving object.

3. Multistage Integration Model

Based on the gradient constraint equation (2.6a), we developed a "multistage integration" model for the terrain shape and egomotion estimates as shown in figure 3.1. In the model, 2D image points $\{I_i\}$ are first processed in an image

processor to generate image measurements $\{\hat{I}_i, \hat{h}_{i\Omega}, \hat{h}_{iv}\}$. Computational operations performed on $\{I_i\}$ at this stage require no knowledge about the observer's motion state nor about the terrain shape, assume no relationship among the image points (e.g., feature relationships), and need not be conducted at all image points. The state estimator and terrain estimator then work in parallel as shown: the state estimator combines the current image measurements with prior state estimates $(\hat{\Omega}, \hat{u}_v)$ and terrain shape estimates (\hat{c}, τ_i) to generate an updated state estimate; likewise, the terrain estimator combines the current image measurements with prior state estimates $(\hat{\Omega}, \hat{u}_v)$ and terrain shape estimates (\hat{c}, τ_i) to generate an updated shape estimate, defined by the terrain model coefficient vector \hat{c} . We now describe these three components in greater detail.

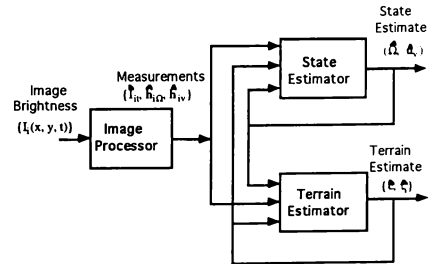


Figure 3.1: Multistage Integration Model

3.1 Image Processor

At a sampling instant, each image frame is defined as a set of intensity measurements $\{I_i\}$, from which the image gradients are generated using the following three-point formulas:

$$\hat{I}_{ix} = \frac{I(\delta_{ix} + \Delta\delta_x, \delta_{iy}, t - \Delta t) - I(\delta_{ix} - \Delta\delta_x, \delta_{iy}, t - \Delta t)}{2\Delta\delta_x} \quad (3.1 \text{ a,b,c})$$

$$\hat{I}_{iy} = \frac{I(\delta_{ix}, \delta_{iy} + \Delta\delta_y, t - \Delta t) - I(\delta_{ix}, \delta_{iy} - \Delta\delta_y, t - \Delta t)}{2\Delta\delta_y}$$

$$\hat{I}_{it} = \frac{I(\delta_{ix}, \delta_{iy}, t) - I(\delta_{ix}, \delta_{iy}, t - 2\Delta t)}{2\Delta t}$$

where Δt is the temporal sampling time, and $\Delta\delta_x$ and $\Delta\delta_y$ are the x and y pixel dimensions. The three-point formula provides higher estimation accuracy than the standard two-point formula ($O(\Delta x)^2$ versus $O(\Delta x)$). Moreover, since the temporal and spatial gradients are estimated at the same (delayed) point in time, no phase shift is introduced. After obtaining \hat{I}_{ix} and \hat{I}_{iy} , the measurement sets $\mathbf{h}_{i\Omega}$ and \mathbf{h}_{iV} are computed using equations (2.6b) and (2.6c). Note that the only source of error in computing $\mathbf{h}_{i\Omega}$ and \mathbf{h}_{iV} is due to the gradient computation since $(\delta_{ix}, \delta_{iy})$ is known precisely. In other words, the accuracy and reliability of the image processor is entirely determined by the image gradient computations.

3.2 Terrain Estimator

Let $\mathbf{P} = (x, y, z)^T$ be a point on the terrain. Normalizing \mathbf{P} by observer speed $|\mathbf{V}|$, we have the impact time vector of (2.2b) given by $\boldsymbol{\tau} = \mathbf{P}/|\mathbf{V}| = (\tau_x, \tau_y, \tau_z)^T$. Let the terrain shape be modeled by an n -th order polynomial, which can approximate any terrain shape to any desired accuracy if we use a high enough order polynomial and the correct coefficients. In particular, let the surface normal impact time be modeled by the following polynomial function:

$$\tau_z = \mathbf{c}^T \mathbf{f} \quad (3.2)$$

where $\mathbf{f} = (1, \tau_x, \tau_y, \dots, \tau_x^n, \tau_x^{n-1}\tau_y, \dots, \tau_y^n)^T$, an n -th order polynomial vector function of the terrain's x - y impact time coordinates, and where the coefficient vector \mathbf{c} is given by

$$\mathbf{c} = (c_0, c_1, c_2, \dots, c_{n^2+n/2})^T \quad (3.3)$$

For the i th point \mathbf{P}_i on the terrain, the corresponding impact time triplet $(\tau_{ix}, \tau_{iy}, \tau_{iz})$ maps to the image point \mathbf{P}_i defined by $\delta_i = (\delta_{ix}, \delta_{iy})$, where $\tau_{ix} = \tau_{iz}\delta_{ix}$ and $\tau_{iy} = \tau_{iz}\delta_{iy}$. Substituting $(\tau_{iz}\delta_{ix}, \tau_{iz}\delta_{iy}, \tau_{iz})$ for the impact time (τ_x, τ_y, τ_z) in \mathbf{f} of (3.2) and representing the in-plane normal impact time τ_{iz} as a function of τ_{ix} and τ_{iy} , we can then obtain an alternative expression for the terrain model using the in-plane image coordinates via:

$$\frac{1}{\tau_{iz}} = \mathbf{a}^T \mathbf{g}_i \quad (3.4)$$

where $\mathbf{g} = (1, \delta_{ix}, \delta_{iy}, \dots, \delta_{ix}^n, \delta_{ix}^{n-1}\delta_{iy}, \dots, \delta_{iy}^n)^T$, a polynomial vector function of the in-plane image coordinates τ_{ix} and τ_{iy} . The constant coefficient vector \mathbf{a} is related to \mathbf{c} via a simple transformation²⁵.

Assume now that the observer has generated measurements $(\hat{I}_{it}, \hat{\mathbf{h}}_{i\Omega}, \hat{\mathbf{h}}_{iV})$, and has available the state estimates $(\hat{\Omega}_i^T, \hat{\mathbf{u}}_i^T)$. Using the gradient constraint equation of (2.5), an inverse impact time map $\langle 1/\hat{\tau}_{iz} \rangle$ can be generated by

$$\frac{1}{\hat{\tau}_{iz}} = \frac{\hat{\mathbf{I}}_{it} - \hat{\mathbf{h}}_{i\Omega}^T \hat{\mathbf{u}}_i}{\hat{\mathbf{h}}_{iV}^T \hat{\mathbf{u}}_i} \quad (3.5)$$

The terrain estimation problem is then one of determining the set of model coefficients \mathbf{a} that minimize the cost function J , given by

$$J = \sum_{i=1}^N \left(\frac{\left(\frac{1}{\hat{\tau}_i} - \mathbf{g}_i^T \mathbf{a} \right)^2}{\sigma_i} \right) + (\mathbf{a} - \mathbf{a}_{old})^T \mathbf{P}_{old}^{-1} (\mathbf{a} - \mathbf{a}_{old}) \quad (3.6)$$

where \mathbf{P}_{old} is the error covariance of the previous model estimates and σ_i is the error variance of the image intensity at the i th image point. Standard recursive least-squares techniques²⁶ can then be used to obtain the optimal coefficient and covariance estimates in accordance with:

$$\hat{\mathbf{a}}_i = \hat{\mathbf{a}}_{i-1} + \frac{\mathbf{P}_{i-1} \mathbf{g}_{i-1}}{\sigma_i + \mathbf{g}_{i-1}^T \mathbf{P}_{i-1} \mathbf{g}_{i-1}} \left[\frac{1}{\hat{\tau}_i} - \mathbf{g}_{i-1}^T \hat{\mathbf{a}}_{i-1} \right], \mathbf{P}_i = \mathbf{P}_{i-1} - \frac{\mathbf{P}_{i-1} \mathbf{g}_{i-1} \mathbf{g}_{i-1}^T \mathbf{P}_{i-1}}{\sigma_i + \mathbf{g}_{i-1}^T \mathbf{P}_{i-1} \mathbf{g}_{i-1}} \quad (3.7 \text{ a,b})$$

where $\hat{\mathbf{a}}_0 = \hat{\mathbf{a}}_{old}$, and $\mathbf{P}_0 = \mathbf{P}_{old}$.

As illustrated in figure 3.2, the terrain estimation process is determined by two key factors: relative motion of the observer with respect to the terrain and a finite field-of-view (FOV) limiting the observer's instantaneous view of the terrain. A rigid terrain surface implies constant coefficients in a terrain-fixed model of the surface. An observer-fixed terrain model of the same surface, however, will yield time-varying coefficients, because of the relative motion of the observer with respect to the terrain surface. Additional coefficient variations with time occur due to the fact that new terrain imagery continuously enters the observer's limited field-of-view. Both factors thus serve to yield a set of continuously varying terrain model coefficients.

A two stage process is adopted in the terrain estimator to accommodate these two factors: The first stage accounts for the relative motion factor by propagating the existing terrain model (from a prior estimate) in time using the current observer state estimates $(\hat{\Omega}_i^T, \hat{\mathbf{u}}_i^T)$. This *dead reckoning* stage corresponds to a pre-measurement *a priori* state estimate in conventional Kalman filtering. The second stage then accounts for the new terrain measurements coming into the observer's field-of-view, by combining the propagated model coefficients with the new measurements. An optimally weighted least squares estimator is used to reflect the relative reliability of the measurements with respect to the propagated coefficients. A

forgetting time factor is used to exponentially unweight the old coefficients in favor of the new measurements, to ensure rapid following of new terrain changes²⁶. This stage corresponds to a post-measurement *a posteriori* state estimate in conventional Kalman filtering.

Figure 3.2 summarizes, in block diagram form, the structure of the terrain estimator. The new measurements, the current vehicle state estimates (obtained from the vehicle state estimator), and the old terrain model coefficient estimates are all inputs to the terrain estimator. Subject to the given terrain model structure, the weighed least squares estimator generates the new estimates for the model coefficients $\hat{\mathbf{c}}$ and impact time maps $\{\hat{\tau}_i\}$. The estimates are then fed back to provide an initial guess for the terrain model at next time frame by propagating the terrain model on the basis of current state estimates.

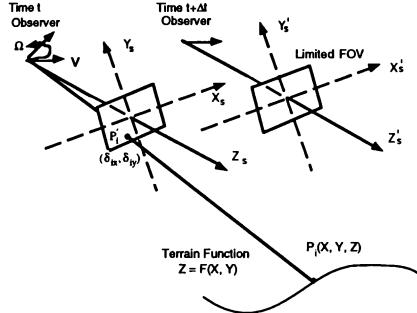


Figure 3.2: Observer Motion Changes

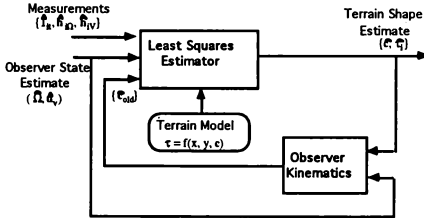


Figure 3.3: Terrain Estimator

3.3 State Estimator

Assume now that the observer has available the new measurements $\{\hat{\tau}_i, \hat{\delta}_x, \hat{\delta}_y\}$ from the image processor and the inverse impact time map $\{1/\hat{\tau}_i\}$ from the terrain estimator. From the gradient constraint equation of (2.5), we have

$$\hat{\mathbf{I}}_{it} = \hat{\mathbf{h}}_{i\Omega}^T \Omega + 1/\hat{\tau}_i \hat{\mathbf{h}}_{iv}^T \mathbf{u}_v + \varepsilon_i \quad (3.8)$$

where ε_i is the error incurred by the image gradient computation and terrain estimation. Motion state estimation is

formulated as a least squares estimation problem that determines $\hat{\Omega}$ and $\hat{\mathbf{u}}_v$ to minimize the following cost function:

$$J(\Omega, \mathbf{u}_v) = \sum_{i=1}^N \varepsilon_i^2 = \sum_{i=1}^N \left(\mathbf{I}_{it} - \hat{\mathbf{h}}_{i\Omega}^T \Omega + \frac{1}{\hat{\tau}_i} \hat{\mathbf{h}}_{iv}^T \mathbf{u}_v \right)^2 \quad (3.9)$$

Denoting $\hat{\mathbf{h}}_i = (\hat{\mathbf{h}}_{i\Omega}, \hat{\mathbf{h}}_{iv}/\hat{\tau}_i)$, the resulting optimal solution can be shown to be:

$$(\hat{\Omega}, \hat{\mathbf{u}}_v) = \left(\sum_{i=1}^N \hat{\mathbf{h}}_i \hat{\mathbf{h}}_i^T \right)^{-1} \sum_{i=1}^N \hat{\mathbf{h}}_i \hat{\mathbf{I}}_{it} \quad (3.10)$$

From the heading vector estimates $\hat{\mathbf{u}}_v$, we can obtain the estimated heading and flight path angles $\hat{\psi}$ and $\hat{\gamma}$ by

$$\hat{\psi} = \tan^{-1}(\hat{u}_{vy}/\hat{u}_{vx}) \quad \hat{\gamma} = \sin^{-1}(\hat{u}_{vz}) \quad (3.11, a, b)$$

The above relationships generate snapshot estimates of the observer's state, based solely on the image measurements, and do not account for any constraints imposed by the observer's dynamics. To properly account for the observer's dynamics, we use conventional Kalman filter techniques to smooth the state estimates. The filter assumes a constant velocity for heading changes and constant orientation changes. Specifically, letting $\theta = (\psi, \gamma)^T$ be the observer's instantaneous heading angle vector, and letting $\Omega = (\omega_x, \omega_y, \omega_z)^T$ be the observer's instantaneous angular velocity vector, the filter assumes

$$\begin{aligned} \dot{\theta}(t) &= 0 \\ \dot{\Omega}(t) &= 0 \end{aligned} \quad (3.12a, b)$$

The filter measurement vector is formed according to

$$\mathbf{y}(t) = (\theta(t), \Omega(t)) \quad (3.12c)$$

and measurements are obtained from the snapshot estimates of (3.10) and (3.11). The filter generates smoothed estimates $(\hat{\Omega}, \hat{\theta})$ which are then used to generate the smoothed estimates $(\hat{\Omega}, \hat{\mathbf{u}}_v)$.

Figure 3.4 summarizes, in block diagram form, the structure of state estimator. The image measurements and the terrain map estimates are first sent to a least squares estimator that generates the snapshot state estimates. Using the observer dynamics model, the estimates are further smoothed by a Kalman filter to produce the final state estimates.

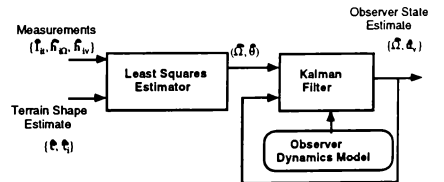


Figure 3.4: State Estimator

3.4 Qualitative Predictions from Model Structure

We now briefly describe some qualitative implications of our modeling approach. We first present an analytical formula that defines image flow error as a function of three key image characteristics: temporal sampling period (or frame time), spatial sampling size (or pixel size), and the intensity quantization level (or gray level resolution). From the relationship, some qualitative implications for human egomotion perception are then discussed.

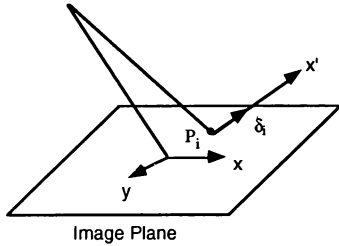


Figure 3.5: Pixel-Centered Image Plane Coordinate System

At a single image point P_i , we can define a new coordinate system $x'y'z'$ as shown in figure 3.5. Here we have located the new system origin at the image point itself (so that $\delta_i = 0$), and aligned the x' -axis with the in-plane image flow vector δ_i . We now define some new variables in the new coordinate system. Along the x' -axis, \hat{L}_i is the estimated spatial gradient, L_i is the dominant wavelength in the image intensity function, $\Delta\delta$ is the effective pixel size, $u_{vx'}$ is the projection of u_v , and v_i is the projection of δ_i , so that $|v_i| = |\delta_i|$. Along the y' direction, $\omega_{y'}$ is the projection of Ω . We also define N_g to be the number of gray levels used in quantizing the image intensity function. It can then be shown that v_i can be expressed as²⁵:

$$v_i = \frac{u_{vx'}}{\tau_i} - \omega_{y'} \quad (3.13)$$

The resulting error in the image flow, due to the temporal, spatial, and intensity quantization factors, is then given by

$$\Delta v_i = \left| \frac{2\pi^2[(v_i\Delta t)^2 - v_i(\Delta\delta)^2]}{3L_i^2} \right| + \left| \frac{v_i\Delta t - \Delta\delta}{6N_g L_i \Delta t \Delta\delta} \right| \quad (3.14a)$$

where the first error term reflects the error caused by the spatial and temporal quantization factors Δt and $\Delta\delta$, and the second error term reflects the error caused by the intensity quantization factor N_g . When $\frac{2\pi^2 v_i \Delta\delta}{L_i} < 1$ and $\frac{2\pi^2 v_i \Delta t}{L_i} < 1$, the *normalized* error formula can be approximated as:

$$\frac{\Delta v_i}{v_i} = \left| \frac{2\pi^2[(v_i\Delta t)^2 - (\Delta\delta)^2]}{3L_i^2} \right| + \left| \frac{v_i\Delta t - \Delta\delta}{v_i N_g L_i \Delta t \Delta\delta} \right| \quad (3.14b)$$

This error formula can provide general qualitative insights to human egomotion and shape estimation, as we discuss in the following paragraphs.

First, from both (3.13) and (3.14), we see that the range-speed ratio has the potential to strongly influence the accuracy of egomotion and shape estimation. This can be seen by first noticing from (3.13) that the flow rate v_i is inversely proportional to the range-speed ratio (here denoted by $\tau_i = \rho_i / |V|$) and then noticing from (3.14b) the dependence of flow estimation accuracy on flow rate itself. With low flow rates (such as occur with long range points near the horizon), we have v_i close to zero, so that the second term of (3.14b) can grow unboundedly large. With high flow rates (such as occur with close range points during low altitude high speed flight) we see that the first error term will scale directly with high flow rate and thus lead to large flow rate errors. Assuming that the flow rate errors of (3.14) directly influence *downstream* state and terrain shape estimation accuracy, it then follows that the range-speed ratio τ_i should have a strong influence on overall egomotion and terrain shape estimation accuracy. This dependency has been the subject of several human egomotion and shape estimation experimental studies (e.g. Eby², Grunwald and Kohn⁴).

A second observation concerns the potential effects of surface texture on expected egomotion and terrain shape estimation accuracy. From (3.14), we see that when the estimated image gradient \hat{L}_i is near zero, the second term can grow without limit. This might occur, for example when long wavelength patterns dominate the *decoration* of the viewed terrain surface, so that the computed spatial gradients are small. We also see that short wavelength patterns (such as obtained with high frequency terrain texturing) can also lead to large errors, since a small L_i leads to unbounded error in the first term of (3.14).

From these observations, it is clear that different regions of the image plane will be characterized by different levels of reliability, in terms of image flow accuracy. It would seem reasonable, then, that any effective egomotion estimator would take this relative reliability into account, either by discounting *high error* image points, or *unweighting* them in a weighted least squares scheme, or use some other technique to guarantee that the high reliability points drive the estimates.

We also see from (3.14) that a judicious choice of temporal and/or spatial sampling steps can minimize the error contributed by both terms. In particular, if we could somehow choose the temporal and spatial sampling steps to satisfy the equation

$$v_i \Delta t - \Delta\delta = 0 \quad (3.15)$$

we could then drive the numerators of both error terms to zero, and thus null the flow error and downstream egomotion estimation error. One means of satisfying (3.15) would be to choose pixel size $\Delta\delta$ at each pixel proportional to flow rate v_i . Thus, in straight and level flight over flat terrain, one might choose to place small pixels on the aimpoint and horizon line, and choose proportionally larger pixels in the visual periphery. The human can accomplish this fairly simply by foveating the aimpoint and relying on the fact that the

visual field receptors increase in size with foveal eccentricity. Clearly, an analogous adaptive temporal/spatial sampling strategy could be used with great effectiveness in comparable computer visual applications.

4. Experimental and Model Results

We now compare results generated by our model with some results obtained in two experiments conducted by Warren³ and Grunwald and Kohn⁴. Detailed explanations of the experiments are given in the respective references.

4.1 Simulation Overview

Figure 4.1 provides an overview block diagram of the engineering simulation environment using computer generated imagery (CGI). Three modules drive the simulation: TERRAIN, OBSERVER, and SENSOR. The TERRAIN module defines the texture pattern on the ground plane, to be imaged by the sensor/observer. The OBSERVER module simulates the kinematics involved in the observer's flying over the ground terrain. The module generates the terrain-relative observer states of position (R), velocity (V), attitude (A), and angular velocity (Ω), and updates them every simulation time step. The SENSOR module simulates the basic geometric attributes of the imaging sensor/observer. It specifies the sensor boresight in the vehicle body-axis coordinates (u_s^b), as well as the size of the FOV (Ψ_{FOV} , Θ_{FOV}) and the lateral and vertical pixel count (N_Ψ , N_Θ). This module supports arbitrary boresighting and orientation of the sensor FOV, for general configurations.

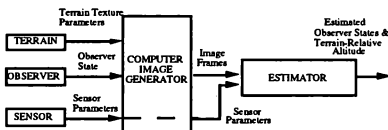


Figure 4.1: Computer Generated Image (CGI) Simulation

The ground texture parameters, observer states, and sensor parameters are then combined by the computer image generator module, to generate a sequence of image frames in true perspective, as seen by the observer, in the overlflight of the simulated terrain. Each image frame is defined by a set of N intensity scalars (I_i), one for each of the N sensor pixels. One image frame is produced every simulation time tick. Processing of the computer generated imagery is then accomplished by our estimation algorithm. Not shown in the diagram are simulation modules for data storage and performance calculations.

4.2 Comparison with Warren's Experiment

In the experiment conducted by Warren³, subjects were "flown" over flat terrain decorated by a uniformly random array of luminous dots. The flight path was straight and level, with an altitude of d units and speed $1.25d$ units/s, where d was the average inter-dot spacing of the terrain texture. The lateral FOV was 53.1 deg, and the vertical FOV was 26.5 deg below the displayed horizon. After viewing the display, the subjects were asked to indicate their lateral heading aim point, and were

able to do so with an accuracy of 1.5 deg standard deviation (across subjects).

To simulate the experimental environment, a sequence of computer images were generated by using the OBSERVER, SENSOR, and TERRAIN simulation modules illustrated in figure 4.1. The simulated vehicle flew straight-and-level at an altitude of 100 feet above the terrain, with zero body attitude rates. The frame time between images was set as 0.033 s (30 Hz) for consistency with our analysis of the Grunwald and Kohn⁴ experiment (see below). The terrain scene was approximated by a two-dimensional sum-of-sines spatial texture pattern, having a fundamental wavelength of 100 ft, with coefficients chosen to approximate the luminous dot pattern used in the original experiment. The image plane was set normal to the observer's velocity vector with a zero depression angle relative to the terrain. The sensor field-of-view (FOV) was set to 53.1° laterally by 53.1° vertically, and the pixel count was set at 100 laterally by 100 vertically, for a total of 10,000 pixels.

Figure 4.2 shows one image frame generated under the above conditions. The upper half of the figure is sky, and has a neutral gray level. The lower portion below the horizon shows the array of luminous dots, which naturally recedes to the horizon. The effects of pixellation are clearly evident.

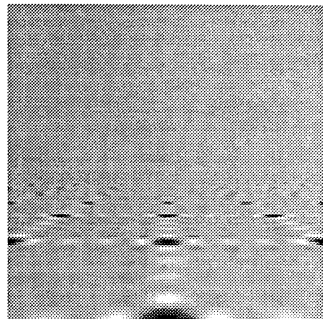


Figure 4.2: Computer Generated Terrain Image used for Simulating Warren³ Experiment

In the simulation, a sequence of these images are generated, and then processed, frame by frame, using the previously described algorithm to obtain estimates of the observer state and terrain parameters. Figure 4.3 illustrates a typical resulting time history of the heading error, computed as the difference between the actual and estimated heading angle defined earlier by (3.11a). The thin line shows estimation error generated by the *snapshot* least squares estimator of figure 3.4. The high frequency content of this signal directly reflects the frame-by-frame variation in the image. The thick line shows estimation error generated by the Kalman filter of figure 3.4, which provides smoothing using the vehicle dynamics. As can be seen, the algorithm provides excellent estimation of the observer's heading by effectively exploiting the available temporal and spatial visual information. It is clear, however, that this algorithm's performance far exceeds that achievable by the human under similar conditions.

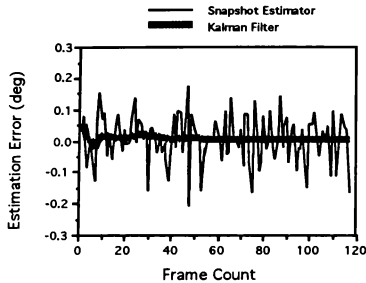


Figure 4.3: Heading Estimation Error Time History: Zero Image Noise

To model human performance under these conditions, we represented human visual perceptual limitations by adding zero mean Gaussian random noise directly to the image. Figure 4.4 illustrates the resulting heading error estimation time history with a 10% image noise level; that is with image noise standard deviation set equal to 10% of the overall grey level of the image. Clearly, this perceptual noise injection significantly increases the magnitude of the *snapshot* or least squares estimate (note scale differences), as well as that of the Kalman filter estimate.

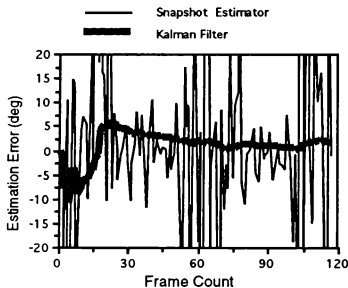


Figure 4.4: Heading Estimation Error Time History: 10% Image Noise

To assess human estimation performance under different perceptual levels, Monte Carlo simulations were run to generate statistics of the resulting estimation error. For each run, we computed the root mean squared (RMS) estimation error over a response time window, starting from 1.5 sec following stimulus presentation (assuming subjects needed at least 1.5 sec to respond (Warren³) and running to 4.0 sec following stimulus presentation (when model response suggests no further improvement in estimation error). We then computed the mean and standard deviation of the RMS estimation error across 30 Monte Carlo simulations of the same experimental condition. This process was then replicated for several different image noise levels.

Figure 4.5 shows mean (triangles) and standard deviation (error bars) of the RMS estimation error as predicted by the model for three different image noise levels: 5%, 10%, and 20%. We see a clear monotonic trend in error with perceptual image noise level. We have also indicated on the figure the 1.5 deg estimation error obtained under the experiment conducted by Warren³, and have indicated how this performance level can be “explained” as an 8% perceptual image noise level, within the context of the model we have presented here.

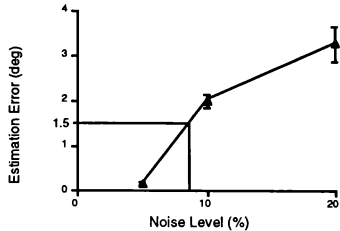


Figure 4.5: Estimation Error vs. Image Noise Level

4.3 Comparison with Grunwald/Kohn Experiment

In one portion of the experiment conducted by Grunwald and Kohn⁴, a subject was “flown” over flat terrain decorated by a field of rectangular patches placed in a cross-grid pattern. As with Warren’s experiment, the flight path was straight and level and the image plane was normal to the observer’s velocity vector. The field of view was 102.7 deg horizontal and 90 deg vertical. Observer altitude was set at d units above the ground plane. The terrain was visible from a distance $1d$ until $15d$ units from the observer. The average length and width of the viewed rectangles was $0.385d$, and the average distance between their centers was $0.625d$. In each trial, the lateral flight path deviated from the viewing axis by the sidestep angle β , which was chosen from a uniformly distributed random set ranging from -45 to $+45$ deg. The subject’s task was to place a marker on the estimated aimpoint. The subject was given a maximum of 8 sec to respond, after which the run was terminated. In the experiment, different velocity-to-height ratios were used, ranging from 0.25 to $4s^{-1}$.

In our simulation of this experiment, the simulated vehicle also flies straight-and-level at an altitude of 100 feet above the terrain, with zero body attitude rates. The frame time between images was set as 0.033 s (30 Hz), which is the screen update frequency of the Silicon Graphics IRIS 4D 50/GT workstation used in the experiment. The terrain scene was approximated by a two-dimensional sum-of-sizes texture pattern, of regularly distributed square patches of size 38.5 ft, spaced 77.0 ft apart. The image plane was set normal to the observer’s velocity vector. The sensor field-of-view (FOV) was set to 90° laterally by 90° vertically, and the pixel count was set at 100 laterally by 100 vertically, for a total of 10,000 pixels.

Figure 4.6 shows one image frame generated under the above conditions. Figure 4.7 shows the corresponding image temporal gradients, which comprise one part of the

image measurement set generated by the image processor (recall figure 2.3).

Figure 4.8 illustrates a typical time history of the heading error, obtained during our Monte Carlo analysis of this experiment, using a 15% image noise level to simulate human perceptual limitations. As before, the thin line shows the output of the snapshot least squares estimator (before filtering) and the thick line shows the output of the Kalman filter.

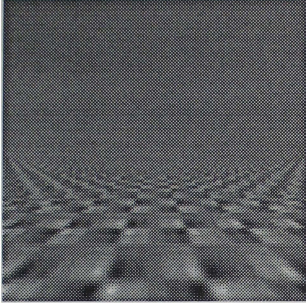


Figure 4.6: Computer Generated Terrain Image used for Simulating Grunwald and Kohn⁴ Experiment

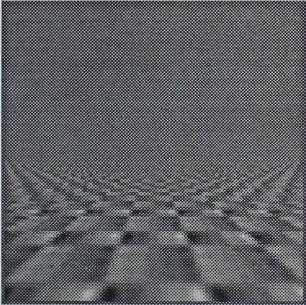


Figure 4.7: Corresponding Image Temporal Gradients for Grunwald and Kohn⁴ Simulation

As with our analysis of the experiment by Warren³, we varied the image noise level to find a best match to the experimental data, across all conditions tested. In this case, we found a 15% image noise level to provide the best overall match. Figure 4.9 shows the resulting RMS estimation error in aimpoint, as a function of velocity-height ratio. Model predictions are indicated by the shaded region, defined as the plus/minus one-sigma envelope about the expected RMS error. Experimental data points are indicated by the solid circles. Note that the data shows a clear trend in which errors vary inversely with velocity-to-height ratio, with the lower flow rates yielding higher errors. This trend is replicated by the model results, as we expect from our previous discussion on the effect of extreme (in this case, very low) flow rates.

Although not covered in this experiment, we would expect a gradual increase in aimpoint estimation error, as flow rates are increased, so that an overall U-shaped function is obtained.

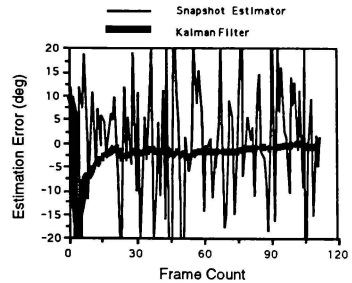


Figure 4.8: Heading Estimation Error Time History: 15% Image Noise

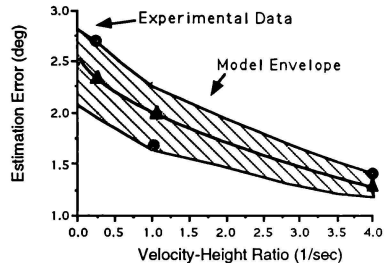


Figure 4.9: Heading Estimation Error vs. Velocity Height Ratio: Data vs. Model

5. Conclusion

We have described a multi-stage integration model of human egomotion perception, based on our work in computer vision systems. The model differs from conventional linear architecture approaches, in which dynamically changing image are used to generate a corresponding optical flow-field, which in turn is used to generate egomotion state estimates, which in turn are used to generate relative depth/shape estimates. In contrast, our model employs multistage integration. Here the image is first processed to generate the basic spatial and temporal image gradients that represent the image motion changes. The information is then directly sent to an egomotion state estimator and a depth/shape estimator. The state estimator uses the image gradient information in combination with a depth/shape estimate of the viewed surface to generate updated state estimates. These state estimates and the spatiotemporal gradient information in turn drive the depth/shape estimator to generate updated depth/shape estimates, which in turn loop back to the state estimator. This feedback loop between state and shape estimators provides us with the desired multistage

integration, and frees us from the artificial constraints imposed by conventional two- or three-stage linear architectures.

There are several qualitative implications of this model for human egomotion perception. On the basis of an explicit functional relationship, we can see how key factors of range-speed ratio (or global flow rate), dominant image wavelength, image resolution, and image quantization level all serve to drive the error in image flow and resulting egomotion estimation accuracy. Both high and low flow rates can give rise to large errors. Other sources of error are high frequency texturing of the terrain surface, near-zero image gradients, and coarse quantization in gray level. The model also points a basis for designing optimal spatiotemporal sampling strategies for using the information available in the visual field: by rejecting and avoiding low-reliability regions in the visual field, high reliability egomotion estimates can be generated. Finally, the model provides use with a means of simulating a variety of egomotion geometry and visual worlds, so that their impact on egomotion perception can be studied in a common context, across a variety of experimental conditions.

This simulation capability allowed us to analyze several earlier experimental studies of human egomotion perception. For the two reported on here, we were able to show how simple additive image noise can be used to model the human's inability to generate accurate spatiotemporal gradients, a shortcoming which leads to corresponding inaccuracy in estimates of egomotion state. Specifically, we showed how, within the model context, an assumed 10% to 15% image noise level leads directly to expected egomotion aimpoint accuracies on the order of 1.5 deg³, and further, how these aimpoint accuracies can be expected to co-vary with velocity-to-height ratios (flow rate)⁴, which change by over a factor of 10. Although more model validation is clearly needed, these initial matches between model and data are very encouraging.

The utility of this model should extend beyond the unification of a number of separate empirical studies, because of its inherent predictive simulation capabilities. In particular, we see considerable potential for use in predicting human egomotion perception as a function of the visual cueing environment of the modern flight simulator. Used as a tool for predicting and comparing egomotion perception in both the flight and simulator environments, the model could provide a sound foundation for evaluating the perceptual fidelity of a simulator, its visual realism, and ultimately, its effectiveness as a training system.

6. Acknowledgement

This work was performed under USAF Contract F41624-92-C-6007 with the Armstrong Laboratory, Human Services Division, Crew Systems Directorate at Wright-Patterson AFB, OH. Technical monitor for the contract is Dr. Rik Warren.

7. References

- 1 Gibson, J.J. 1950. *The Perception of the Visual World*. Houghton Mifflin, Boston.
- 2 Gibson, J.J. 1966. *The Senses Considered as Perceptual Systems*. Houghton-Mifflin, Boston.
- 3 Warren, R. 1976. "The Perception of Egomotion." *J. Exp. Psych: Human Perception and Performance*, Vol. 2: pp. 448-456.
- 4 Grunwald, A.J. and Kohn, S. 1993. "Flight-Path Estimation in Passive Low-Altitude Flight by Visual Cues." *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 2, March/April: pp. 363-370.
- 5 Eby, D.W. 1992. "The Spatial and Temporal Characteristics of Perceiving 3-D Structure Motion." *Perception & Psychophysics*, Vol. 51, No. 2: pp. 163-178.
- 6 Warren, W.H. and Kurtz, K.J. 1992. "The Role of Central and Peripheral Vision in Perceiving the Direction of Self Motion." *Perception and Psychophysics*, Vol. 51, No. 5: pp. 443-454.
- 7 Crowell, J.A. and Banks, M.S. 1993. "Perceiving Heading with Different Retinal Regions and Types of Optic Flow." *Perception and Psychophysics*, Vol. 53, No. 3: pp. 325-337.
- 8 Weng, J., Huang, T.S. and Ahuja, N. 1992. "Motion and Structure from Line Correspondences: Closed-Form Solution, Uniqueness, and Optimization." *IEEE Transactions on PAMI*, Vol. 14, No. 3: pp. 318-336.
- 9 Watson, A.B. and Ahumada, A.J. 1983. *A Look at Motion in the Frequency Domain*. NASA. 84352 (April 1983).
- 10 Watson, A.B. and Ahumada, A.J. 1985. "Model of Human Visual-Motion Sensing." *Journal of Optical Society of America A*, (February): 322-342.
- 11 Heeger, D.J. 1987. "Model for the Extraction of Image Flow." *Journal of the Optical Society of American A*, (August): 1455-1471.
- 12 Horn, B.K.P. and Schunck, B.G. 1984. "Determining Optical Flow." *Artificial Intelligence*, Vol. 72: 1671-1686.
- 13 Schunck, B.G. 1983. *Motion Segmentation and Estimation*. Sc.D Thesis. Massachusetts Institute of Technology.
- 14 Schunck, B.G. 1989. "Image Flow Segmentation and Estimation by Constraint Line Clustering." *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 10: pp. 1010-1027.
- 15 De Micheli, E., Torre, V. and Uras, S. 1993. "The Accuracy of the Computation of Optical Flow and the Recovery of Motion Parameters." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 5, May: pp. 434-447.
- 16 Zacharias, G.L., Caglayan, A.K. and Sinacori, J.B. 1985a. "A Model for Visual Flow-Field Cueing and Self-Motion Estimation." *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 3: 385-389.
- 17 Zacharias, G.L., Caglayan, A.K. and Sinacori, J.B. 1985b. "A Visual Cueing Model for Terrain-Following Applications." *AIAA Journal of Guidance*, Vol. 8, No. 2: 201-207.

- 18 Broida, T.J. and Chellappa, R. 1986. "Estimation of Object Motion Parameters from Noisy Images." *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 8: 90-99.
- 19 Merhav, S.J. and Bresler, Y. 1986. "On-Line Vehicle Motion Estimation from Visual Terrain Information, Parts I and II." *IEEE Trans. on Aerospace and Electronic Systems*, : 583-603.
- 20 Sridhar, B. and Phatak, A.V. 1988. "Simulation and Analysis of Image-Based Navigation System for Rotorcraft Low-Altitude Flight." *AHS National Specialists' Meeting Automation Applications of Rotorcraft*. Atlanta, GA.
- 21 Sridhar, B., Cheng, V.H.L. and Phatak, A.V. 1989. "Kalman Filter Based Range Estimation for Autonomous Navigation Using Imaging Sensors." *XIth IFAC Symposium on Automatic Control in Aerospace (July)*. Tsukuba, Japan.
- 22 Young, G.J. and Chellappa, R. 1990. "3-D Motion Estimation Using a Sequence of Noisy Stereo Images: Models, Estimation, and Uniqueness Results." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 8: 735-759.
- 23 Zacharias, G.L. and Levison, W.H. 1980. *Development of a Model for the Use of Extra-Cockpit Visual Cues*. Bolt Beranek and Newman Inc. 4562 (December).
- 24 Zacharias, G.L., Miao, A.X. and Riley, E.W. 1992. *Passive Sensor Altitude and Terrain Awareness System*. Charles River Analytics Inc. R91071 (February).
- 25 Miao, A.X. and Zacharias, G.L. 1993. *Motion and Terrain Parameter Estimation from Image Data*. Charles River Analytics Inc. Technical Memo TM92191 (June).
- 26 Goodwin, G.C. and Sin, K.S. 1984. *Adaptive Filtering Prediction and Control*. Eaglewood Cliffs, NJ: Prentice-Hall, Inc.

VISUAL WEATHER SIMULATION USING METEOROLOGICAL DATABASES

Bruce Montag*

Manager, System Simulation and Modeling
 Southwest Research Institute
 San Antonio, Texas 78228

Abstract

This paper presents the results of an internal research program at Southwest Research Institute for the development of a visual weather simulation and modeling system for flight training applications. This weather simulation system concept, first presented at the 1990 AIAA Flight Simulation Technology Conference, provides the means to correlate and synchronize all weather-related cues presented to the aircrew. The approach provides for direct correlation between out-the-window visual weather scenes, weather-processing sensor avionics displays, and aircraft handling qualities through the use of a unified meteorological database. By ensuring dynamic weather cue correlation across all simulator subsystems, this technique enables simulator instruction in weather-related flying to be highly transferable to in-flight, mission-oriented situations. This research effort produced a unique method for generating real-time fly-through weather conditions on a polygon-based visual system. The model architecture also supports sensor simulation for driving forward-look wind profiling models and cockpit displays. Since the weather model is driven by gridded-field, digital meteorological data, aircrews can learn and practice line-oriented weather avoidance skills in a realistic simulated weather environment.

Introduction

In the commercial aviation world, microburst windshear, convective weather, clear air turbulence, and wake vortices are widely recognized weather hazards to flight safety. These phenomena have a direct cost impact on flight operations due to increased stress on airframes, flight delays, increased fuel costs, and potential passenger lawsuits. Doppler radar and other sensing devices capable of remotely detecting hazardous atmospheric conditions are emerging, but aircrew judgment will always be

heavily relied upon for weather interpretation and avoidance. Pilot training in adverse weather detection and avoidance methods is currently limited due to the lack of realistic weather modeling capability in flight simulators. Providing tools and training for pilots to avoid flight into hazardous weather situations is a pressing challenge within the aviation industry.

In the military world, weather plays a key role in mission planning, weapon and sensor system usage, and mission tactics for military operations. Current weapon system trainers (WSTs) and simulation networks are limited in their ability to provide a realistic portrayal of weather conditions that are applicable to mission operations.

Southwest Research Institute (SwRI) conducted an internal research program to develop an affordable, achievable technique for modeling and visualizing real-world weather effects for flight simulator application. Visual effects were chosen for research since this area was judged to be the most difficult and featured the highest payoff for simulator training. This capability is needed to improve transfer of training metrics for simulator-based hazardous weather avoidance as well as to introduce more realistic weather for mission rehearsal training. The SwRI-developed weather modeling approach, known as the Weather Environment Simulation Technology (WEST) method, allows full simulation and visual display of very large atmospheric data sets containing fine-grained atmospheric conditions associated with convective and turbulent weather, as well as visual obscuration such as smoke and dust plumes. WEST enables the simulator user to see and react to weather conditions visually in addition to observing correlated cues from sensors such as radar, FLIR (forward-looking infrared), windshear detection, and flight subsystems. WEST allows the user to visually observe, "fly through," and interact with recorded atmospheric data at real-time speeds using commercial off-the-shelf hardware.

*Member AIAA

Technical Approach

Advances in computer image-generation (CIG) technology and weather modeling methods provide new capabilities for simulating and visualizing weather effects in four dimensions (space plus time). SwRI evaluated several visualization techniques on high-capacity graphics workstations to determine the best method for flight simulator application. An approach using continuous-level-of-detail textured graphic primitives was selected as the best method, and a frame-based image-generation transform function was developed for constructing and rendering scene primitives from weather database parameters in real time.

The WEST application works with the Silicon Graphics GL library to render images of weather effects through the real-time interpretation of digital weather data structured in a gridded Cartesian format. The WEST approach is compatible with any gridded field weather database structured in a Cartesian format. Both uniform and non-uniform grid formats may be accommodated. WEST allows the user to view this data in either a vector/scalar format or as simulated optical imagery. Weather data parameters for visualization accommodated within WEST include wind direction and magnitude, liquid water content, temperature, pressure, radar reflectivity, and water content type (rain, snow, ice). Digital atmospheric data sets currently supported include the Joint Airport Weather Studies (JAWS) database from the National Center for Atmospheric Research (NCAR) and the Terminal Area Simulation System (TASS) meteorological weather model developed by the NASA Langley Research Center for windshear research.

Meteorological Databases

Sources of remotely sensed atmospheric data have expanded rapidly within the last several years, including ground-based doppler weather radars and several satellite-based earth-observing systems that sense and record large-scale weather conditions. Although these systems are capable of producing voluminous amounts of data, tools for visualizing, analyzing, and applying the data for operational uses requiring four-dimensional display (such as flight simulators) have not kept pace.

Four-dimensional weather databases from NCAR's JAWS project and NASA's TASS model served as the baseline weather environments for

modeling and visualization. These databases are typical of the digital atmospheric data that may be utilized for flight simulator application with the WEST approach. The NCAR database contains gridded parametric data for wind speed and direction, radar reflectivity, and liquid water content at 200-meter grid intervals within a 16km by 16km volume of airspace. The TASS database is similar and includes additional data describing temperature, pressure, and water content type.

Basis for Research

Current practices for modeling and visualizing four-dimensional weather effects do not provide the necessary visual cues, cue correlation, rendering performance, or environmental modeling features that are required for many applications of digital atmospheric data. Flight simulation is one application domain that exemplifies current limitations in the visual simulation of weather data.

State of the Art and Limitations

Interactive visual simulation of weather data has traditionally been implemented in the form of both two-dimensional (2-D) imagery of the type commonly seen on the evening news, and three-dimensional (3-D) computer-generated imagery. An excellent review of current practices for visualizing meteorological data is provided by Papathomas et al.¹ Most of these commonly applied techniques allow the user to view the spatial distribution of various atmospheric parameters such as temperature or liquid water content occurring within a bounded area of the atmosphere, but they essentially provide only two-dimensional views of four-dimensional (4-D) data. Additional techniques referenced by Hibbard² focus on visualizing meteorological data in three dimensions for non-real-time data interpretation and scientific analysis purposes.

In sophisticated 4-D applications such as the out-the-window image generators used in flight simulators, weather conditions are commonly presented through the use of traditional 3-D computer graphics techniques that are optimized for real-time image generation involving solid surfaces and/or textured surfaces.

Solid Surfaces. Early flight simulator visual systems used solid-surface polygonal objects to represent cloud formations. These visual cloud representations were created manually using a

database modeling system. With the solid-surface approach, clouds are constructed by stretching a tri-mesh-type polygonal surface around the exterior of a cloud. The surface is then colored, shaded, and in some cases assigned a transparency value or material code to provide the cloud's visual appearance. Cloud objects are then grouped together to form weather formations.

The main limitations of solid surfaces for visualizing weather are limited visual realism and the difficulty of correlating visual appearance with digital atmospheric source data used to drive other simulator subsystems. Creating realistic weather effects with solid surfaces requires a lot of hands-on work at the database modeler's station by a highly skilled designer. A mathematical model of weather pattern dynamics must also be developed so that during visualization the individual cloud objects can be dynamically sized, oriented, and positioned. Other significant limitations are that varying resolution models must be developed to accommodate viewing under both far and near distances, and that special visual effects are required to accommodate reduced visibility when inside cloud boundaries. (An example of a modified solid-surface approach (textured ellipsoids) that has demonstrated the ability to produce realistic-appearing cumulus clouds is outlined by Gardner.³)

Textured Surfaces. Current-generation flight simulator visual systems use texture to produce photorealistic weather effects. These "texture maps" allow scanned photographic images to be mapped onto polygon surfaces to provide visual weather conditions. The most common practice is to apply weather texture maps to very large single-polygon "billboards" that are oriented either horizontally (parallel to the ground) or vertically (perpendicular to the ground) to provide the appearance of cloud tops/bottoms or weather formations on the horizon.

The primary weakness of the texture map approach is that, although it is easy to implement with state-of-the-art hardware, it is exceedingly difficult to correlate 2-D photographic texture with 4-D digital atmospheric data. The 2-D textured billboard method cannot support viewpoint-independent viewing, or fly-through viewing—features that are required or desired for many types of atmospheric visualization applications.

Requirements for the Visual Simulation of Weather

An ideal approach for overcoming the limitations of current techniques for visualizing and modeling weather is to render imagery directly from digital source data, rather than manually creating solid models or textured billboards and then attempting to manually correlate the visual effects with source data weather conditions. Digital atmospheric data is becoming increasingly available in the form of 3-D gridded data sets. Many users and applications for digital atmospheric data desire the capability to interactively visualize this data as optical or sensor imagery in four dimensions. The fourth dimension is modeled by time-tagging the 3-D gridded data sets. Essential visual simulation capabilities for rendering weather imagery from gridded data sets have been outlined by the author⁴ and include 3-D gridded-field data handling, independent viewpoint viewing, and fly-through viewing.

Digital Data Handling Capability. In order to render imagery directly from digital atmospheric data, the visual simulation technique must be capable of manipulating and processing 3-D gridded data very rapidly to support real-time scene update rates (typically 30Hz). The data handling technique must accommodate paging between spatial weather data subdivisions or tiles as well as continuous interpolation between adjacent time-stamped data files. This capability is essential for handling very large spatial and temporal data sets.

Independent Viewpoint Capability. A key requirement is that the ideal digital weather visual simulation technique must support totally independent viewpoint capability for scene rendering of the gridded weather data set. This means that the data must be viewable from any position or orientation with correct perspective and aspect, including "inside out" viewing. This capability is essential for flight simulation applications where the viewpoint is almost always located within the boundaries of the 3-D gridded field. In many circumstances for high-altitude weather viewing, the entire data set may be located within the field of view.

Fly-Through Viewing Capability. In addition to independent viewing, the optimal visual simulation technique for 4-D weather display must also support seamless fly-through viewing of the data. The user

should be presented with perspective-correct imagery under all viewing conditions. The visual appearance of the weather effects should automatically compensate for viewing aspect, lighting conditions, and relative motion of the eyepoint and dynamic weather effects data. Many of the key visual simulation needs for generating weather imagery in commercial flight simulators were described by Allsopp⁵ over 14 years ago. Some of these specific desires were to realistically depict specific cloud formations and thunderstorm effects, provide the visual effects of onrushing precipitation and snow, and present the blooming and attenuation effects of strobe/landing lights when flying through clouds.

WEST Approach

The WEST approach is specifically designed to accommodate the real-time visual simulation requirements associated with rendering 4-D weather imagery directly from digital source data. The WEST prototype developed by SwRI⁶ executes on a Silicon Graphics Crimson Reality Engine with two Raster Managers. Figure 1 illustrates the architecture and functional elements of the WEST approach. Major elements include the digital weather source database, a visual preprocessor component, a data handling component, a simulation interface component, a visual weather database, and an image-generation component.

Digital Weather Source Database

WEST is designed to process large-scale gridded data sets containing digital atmospheric data parameters that may be produced from sensor observations or from numerical models. Digital atmospheric physics models such as TASS⁷ and Doppler radar-derived atmospheric observations such as JAWS⁸ data sets provide quantitative, time-varying descriptions of aviation weather conditions.

For the TASS data, each grid point within the model domain contains 11 atmospheric parameters describing the state of the air mass volume for that spatial location at a given point in time. These parameters include north, east, down wind speed, pressure, temperature, liquid water content, ice content, precipitation rate, snowfall, and hail fall. Grid spacing for the microscale TASS model varies between a maximum of 200 meters down to 40 meters within a typical domain size of 500 cubic miles (10mi × 10mi × 5mi).

Sensor-derived datasets are typically more mesoscale-oriented and contain grid spacing on the order of 500 to 1000 meters, and cover thousands of cubic miles of atmosphere. Grid domains for these data sets can extend to 2048 × 2048 × 15 elements. Weather parameters typically supported include temperature, wind speed, pressure, water content, and water type.

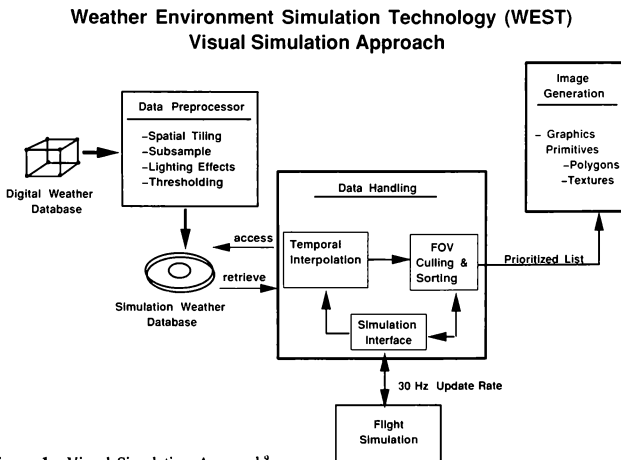


Figure 1. Visual Simulation Approach⁹

Visual Preprocessor Component

Preprocessing methods were implemented within WEST for reading in gridded weather database files and reformatting the data for real-time simulation. Data reformatting includes tiling the data into spatial subdivisions known as cells, and preprocessing/compressing the source data to support image generation. Preprocessing operations include spatial subsampling, data thresholding, coordinate transformation into north-east-down coordinates, liquid water content calculation, graphic primitive assignment, apparent lighting calculation, and spatial dithering. Graphic primitives are assigned to weather data elements according to a parametric data look-up table. These graphic primitives are stored in a library format and include textures as well as polygon primitives. Lighting effects are precomputed for each graphic primitive based on sun position as determined by a time-of-day, day-of-year look-up table. A graphical user interface allows the user to interactively control preprocessing functions and evaluate the visual appearance of the resulting visual weather database.

Visual Weather Database

The visual weather database is the run time database that has been reformatted, tiled, and compressed to support real-time visual simulation. Visual weather parameters maintained within the run time database on a per-element basis include wind vector, texture assignment, spatial extent, color, illumination, and transparency. This database consists of spatial and temporal linked files that are retrieved from disk and then loaded into memory as needed during visualization.

Flight Simulation Interface Component

The flight simulation interface component receives instantaneous aerodynamic environment parameters as a function of viewing platform (aircraft) position. Weather data elements surrounding the viewpoint position are interpolated tri-linearly to determine the instantaneous atmospheric environment including the net aerodynamic effects acting on the viewing platform. These effects include north, east, down wind speed components, time rate of change of these components, and ambient temperature and pressure at the aircraft. The simulation interface component receives the viewpoint and viewing orientation from the flight simulation host computer and provides this data synchronously to the image-generation component. This capability allows WEST

to be integrated with flight simulators to assure direct correlation between visual weather imagery and aircraft dynamic modeling functions.

Data Handling Component (Weather Generator)

The weather-generation component performs the operations necessary to process and format the volumetric weather data visible in the instantaneous display field of view. The weather generation component is scheduled by a frame-based executive to assure periodic weather data update rates and synchronization with the flight simulation and image-generation functions. The weather-generation component controls weather scene management and building of the weather display list that is converted to immediate mode display data by the image-generator hardware. Weather data handling operations include data retrieval and temporal interpolation, field-of-view (FOV) culling, sorting, and prioritizing of in-FOV weather data, and formatting/distribution of active weather elements to the image-generation component. This component could also format and distribute weather data within a given sensor's scanning field of regard or instantaneous field of view.

Data Retrieval. During data retrieval operations, adjacent-in-time cells within the aircraft's field of regard (FOR) are accessed and continually interpolated according to the aircraft's position, orientation, and mission elapsed time. Temporal interpolation between data sets that are separated by a given time interval is a complex task that is the subject of continuing research and development.

Field-of-View Culling. The field-of-view culling and sorting function accesses the run time visual weather database and produces a visible weather display list for each image-generation frame (1/30th of a second). Several operations are involved within the FOV culling and sorting function. These are FOV culling, FOV sorting, and display list processing. The

visual field of view determines what parts of the weather effects can be viewed from the current viewpoint, viewing orientation, and perspective. The culling procedure eliminates weather data elements from the image-generation process that are not within the instantaneous visual field of view.

Field-of-View Sorting. Individual weather data elements that are determined by the culling function to be within the instantaneous field of view are sorted

in front-to-back order as a function of the viewing direction vector. The sorted weather data elements are then formatted into a prioritized weather display list. This prioritization process is based on the location and depth of the weather data element within the field of view; the idea being to optimize the generation of weather imagery based on the capacity of the image generator (IG).

The greater the capacity of the image generator, then the greater the number of depth bins that can be accommodated. Weather data elements that are closest to the viewpoint have the highest priority as a function of location within the field of view.

The horizontal and vertical fields of view are divided into equally spaced "viewing bins" (equally spaced by projected distance in the viewing plane, not by field-of-view angle). Each viewing bin is then allocated a number of depth bins, based on the capacity of the IG. The depth bins for each two-dimensional viewing bin are for storing (sorted by distance) the closest weather data elements in the visual weather database that are within that field-of-view segment. An additional requirement placed on these depth bins is that the elements closer to the viewpoint are increasing in the magnitude of the associated projected wind vector. Depth bins are filled by summing the area of coverage of weather elements for every viewing bin. A front-to-back sorting through the prioritized visible weather elements is employed so that when the depth bins for a particular field-of-view viewing bin become filled, no further processing is required on sample points within that field-of-view segment.

Once all the bins have been filled or all the visible weather elements have been sorted into these bins, whichever comes first, the display list of weather elements is processed for rendering in an order which exhausts the deepest bin in each of the segments first, and then moves forward through the depth bins. This approach results in assured frame display times without sacrificing image quality since the number of weather elements to be rendered during each frame is fixed, and the display list is assured to contain the most visibly significant weather elements due to the prioritization process.

Image-Generation Component. The image-generation component processes the visible weather element display list that was produced by the weather generator and transforms these weather elements into a prioritized list of polygon vertices and graphic library-specific parameters for immediate mode rendering on the image generator's graphics pipeline.

WEST Applications

Flight Simulation

The WEST method for processing digital atmospheric data in real time is ideally suited for flight simulator applications. Figure 2 illustrates the WEST concept for achieving cue correlation across individual simulator subsystems. With the WEST approach, a weather generator may be integrated within the simulator configuration to process and distribute digital weather data to each atmosphere-sensing simulator subsystem. By this method, weather elements from a unified digital database are distributed to each subsystem (visual, radar, flight) as a function of subsystem parameters (position, orientation, field of regard). Each subsystem then performs the required function for transforming digital weather elements to display cues. The visual system transforms the weather data elements into scene primitives (as demonstrated by WEST), and the radar simulation subsystem incorporates the weather data elements into range bin processing for calculating weather attenuation and backscatter parameters.

Figures 3 and 4 illustrate the visual realism available with the WEST technique. Figure 3 is an out-the-window view generated from NCAR gridded-field weather data and Defense Mapping Agency Digital Terrain Elevation Data (DTED). Figure 4 is a visual weather scene produced from SwRI-generated gridded-field data.

Specific commercial aviation applications that could benefit from the use of meteorological databases include wind/shear, thunderstorm and other types of hazardous weather avoidance training. A unified meteorological database approach could also enhance line-oriented flight training through the true correlation of weather cues.

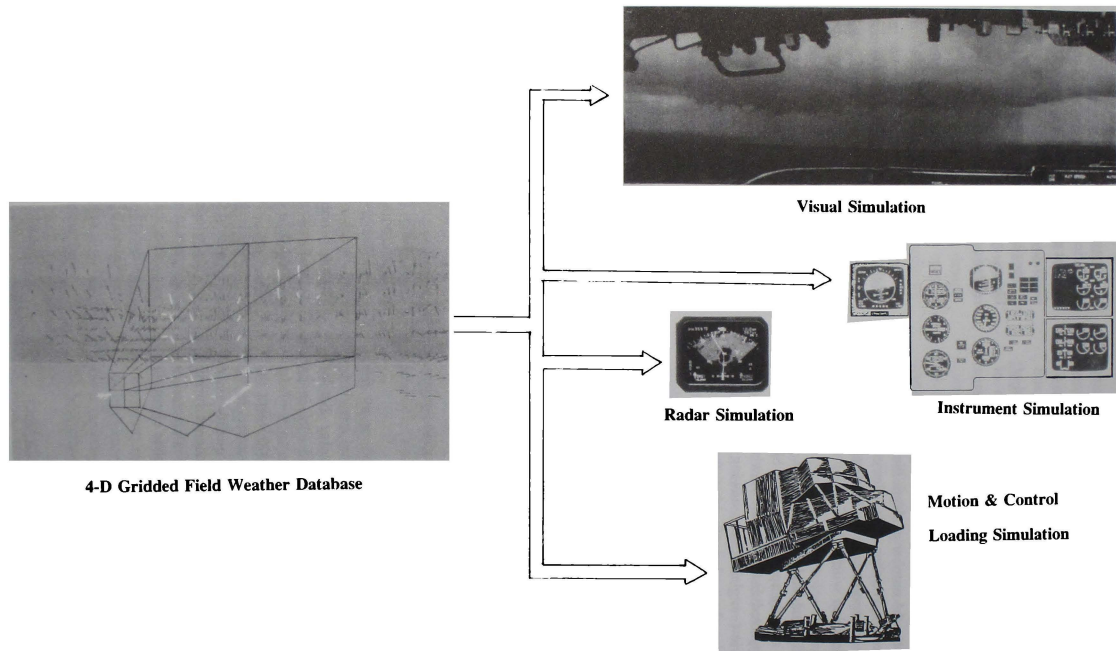


Figure 2. WEST Unified Weather Environment



Figure 3. WEST Computer-generated Weather Scene

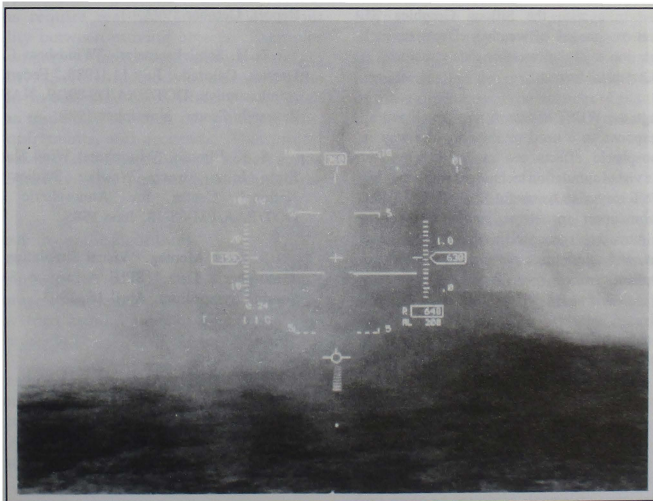


Figure 4. WEST Synthetic Weather Effects

Mission Planning and Mission Rehearsal

Mission planning and mission rehearsal training could become more effective and transferable to operations through the application of the WEST approach for visualizing weather conditions that are expected or forecast for a given mission. Weather environment effects on weapon system employment and combat tactics procedures could be simulated and evaluated by integrating visual weather conditions into the synthetic mission environment that is provided for mission planning systems and tactical preview systems. Given the gridded-field data handling capability of the WEST approach, it is conceivable that satellite-derived weather conditions could be formatted to produce a geospecific, synthetic weather environment for mission rehearsal training. Incorporating satellite weather data for simulator training is a leading area of continued research and development.

Summary

This paper has presented an overview of the SwRI-developed WEST weather simulation approach. This system is capable of providing fly-through visualization of large atmospheric data sets using commercial off-the-shelf hardware. The WEST application works with the Silicon Graphics GL library to render images of weather effects through the interpretation of digital weather data structured in a gridded Cartesian format.

Although the WEST approach was developed by SwRI in response to a need to improve the way in which atmospheric effects are modeled for flight training, the visual simulation techniques implemented within WEST may also be useful for solving related weather simulation problems including sensor simulation modeling, geospecific mission rehearsal applications, and multi-force distributed network training applications.

References

1. T. V. Papathomas et al., "Applications of Computer Graphics to the Visualization of Meteorological Data," Computer Graphics, Volume 22, No. 4, August 1988.
2. W. L. Hibbard, "Computer Generated Imagery for 4-D Meteorological Data," Bulletin of the American Meteorological Society, Volume 67, No. 11, pp. 1362-1369, November 1986.
3. G. Y. Gardner, "Representing Dynamic Cloud Volumes with Fractal Ellipsoids," IMAGE VI Conference Proceedings, pp. 475-483, July 1992.
4. B. C. Montag, "Adverse Weather Simulation Concepts for Safety of Flight Training," Paper No. AIAA-90-3132-CP, AIAA Flight Simulation Technologies Conference, 1990.
5. W. J. Allsopp, "Proposed Advancements in Simulation of Atmospheric Phenomenon for Improved Training," AGARD Conference Proceedings No. 249, Piloted Aircraft Environment Simulation Techniques, pp. 6-11, October 1978.
6. B. C. Montag, "Adverse Weather Visualization Modeling," SwRI IR&D Project 05-9665 Final Report, October 1992.
7. H. Schlickennaier, "Windshear Case Study: Denver, Colorado, July 11, 1988," Federal Aviation Administration, DOT/FAA/DS-89/19, NASA Langley Research Center, November 1989.
8. K. Elmore, "Microburst Wind Shear Models From Joint Airport Weather Studies (JAWS)," National Center for Atmospheric Research, DOT/FAA/PM-85-18, June 1985.
9. B. C. Montag, "Visual Simulation of Digital Atmospheric Data," SPIE Aerospace Science and Sensing Symposium, April 16, 1993.

LINE-OF-SIGHT DETERMINATION IN REAL-TIME SIMULATIONS

Frederick G. Kull Jr.
Senior Systems Engineer
SYRE, a division of SYSCON Services, Inc.
NASA-Ames Research Center, Moffett Field, California

Donald E. Fought, Ph.D.
Group Manager, Fidelity and Technology
SYRE, a division of SYSCON Services, Inc.
NASA-Ames Research Center, Moffett Field, California

Abstract

This paper describes the selection of a method for determining line-of-sight in real-time simulations for the NASA Ames Vertical Motion Simulator (VMS) facility. Five different combinations of terrain representation and line-of-sight determination algorithms were tested. A gridpost terrain format, in conjunction with a Digital Differential Analyzer algorithm, was found to best meet the simulation criteria of high speed, low storage requirements, and accuracy.

Introduction

The VMS is used to perform aeronautical research in the development of flight vehicles. Many of the simulations require real-time information about intervisibility between moving objects. Examples include ownship-to-target and missile-to-target intervisibility. Several methods of calculating the line-of-sight in real-time exist. (Refs. 1 - 4) Each method has trade-offs between computation time, storage requirements, and accuracy. This project was undertaken to determine the method best suited to the VMS facility. The requirements included the following:

- 1) flexibility of the algorithm;
- 2) speed;
- 3) accuracy;
- 4) storage space; and
- 5) low turn around time to generate a new terrain file.

Testing was conducted using a Digital Equipment Corporation VAX 9000-210 running the μ Tau real-time operating system developed at the VMS facility.⁵ An Evans and Sutherland CT5A visual display system was used for image generation.

The evaluation process begins with the description of the terrain file formats used by the various algorithms. The formats were evaluated based on storage requirements and ease of modification. The next section of the paper describes the five line-of-sight determination algorithms that were tested. These algorithms were evaluated on the basis of speed and accuracy. Test results and a review of the selection process are followed by an example of the application of the selected algorithm to a forward-looking sensor.

Terrain File Formats

All of the line-of-sight algorithms tested use some type of data file to model the surrounding terrain, hereafter referred to as a terrain file. Two terrain file formats were tested: 1) storage of terrain height information only at the intersection points of a uniform grid overlaying the visual database, hereafter called the gridpost method; and 2) storage of the coordinates of the vertices of planes in a visual database, hereafter called the edge method.

Gridpost Terrain

The gridpost method consists of a two-dimensional data array representing the terrain height at the intersection points on a uniform inertial grid overlaying the visual database. The array indices are X and Y integer coordinates of the axis system defined by the grid and the terrain height information is stored as inertial Z coordinates in the array. The inertial X and Y coordinates of the lower left post in the grid and the post to post spacing are used to transform inertial positions into the grid axis system. Due to its appearance, such a terrain representation has been called a "bed-of-nails", as shown in Figure 1.

There are two ways to create such a terrain file. The first method involves dropping vertical lines at the

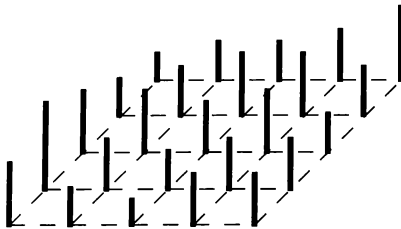


Figure 1 Gridpost Terrain (Bed of Nails)

positions of the gridposts onto the planes stored in a visual database. The second method involves using Digital Terrain Elevation Data (DTED) files supplied by the Defense Mapping Agency (DMA).

In this evaluation, the terrain height information was stored as 2 byte integers to minimize the memory requirement of storing sizable databases. This allows terrain altitudes of plus or minus 32,767 feet with a one foot resolution. The terrain file was read into the executing program at run time.

Edge Terrain

A visual database is made up of many interconnected planes whose vertices are connected by line segments defining the edges of the planes, as shown in Figure 2. The edge terrain method is a representation of each of these planes. The file

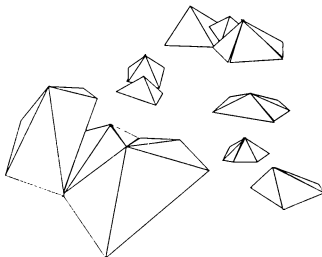


Figure 2 Edge Terrain

contains the X, Y, and Z coordinates of each pair of endpoints of the edge line segments extracted from a visual database. In this study, this information was coded directly into DATA statements.

Comparison

Routines enabling quick extraction of terrain height information can be applied to both terrain methods, resulting in similar turnaround times. Thus, the primary concern when comparing these two methods was computer memory storage requirements.

The edge method requires less storage space than the gridpost method for simple databases comprised of a small number of triangular planes. Most of the edges are concentrated in areas of complex terrain with very few edges in large flat areas. The gridpost method uses the same number of posts independent of the complexity of the terrain. However, all of the new terrain databases at the VMS facility are created using a uniform grid in which the small post spacing produces complex databases with very few large flat areas. The large number of vertices required to represent these databases with the edge method increases the storage requirements.

The existence of obstacles, such as trees and buildings, on the visual database also affects the storage requirements of the edge method. These obstacles are typically located offset from the edges of each database plane. Obstacle heights are easily included in a gridpost terrain file by adding the obstacle heights to the terrain height at appropriate gridposts. If the post spacing is 10 meters, a maximum lateral positional error of plus or minus 5 meters is introduced. However, no additional posts are needed to store obstacle information. With the edge terrain, the vertices of each obstacle must be added to the vertex list, thus increasing storage requirements.

One advantage of the gridpost method is the existence of several different algorithms for determining line-of-sight using this form of terrain information. This means there are more options available when selecting the optimal method for a particular task or environment. Another advantage of this method is that the terrain file corresponds closely to the DTED files supplied by the DMA. DTED files can be used in the VMS environment with very little preprocessing, thus effectively reducing the simulation development time.

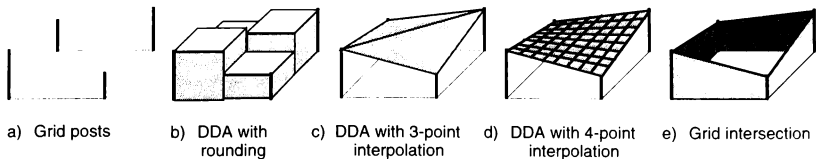


Figure 3 Equivalent Terrain Model Comparison

Line-of-Sight Algorithms

The purpose of any line-of-sight algorithm is to determine if an unobstructed line-of-sight exists between an eyepoint and a target. Each of the algorithms tested begins by creating a Line-Of-Sight (LOS) vector connecting the eyepoint and the target. The terrain height is then compared to the altitude of this LOS vector at locations along the length of the vector. If the terrain is higher than the LOS vector at any one of these locations, the line-of-sight is considered blocked.

The five algorithms tested are as follows:

- 1) Digital Differential Analyzer with rounding;
- 2) Digital Differential Analyzer with three point interpolation;
- 3) Digital Differential Analyzer with four point interpolation,
- 4) grid intersection algorithm employing two point interpolation; and
- 5) intersection of line segments employing the theory of inequalities.

The first four algorithms use the gridpost terrain file, but each defines the terrain between the grid posts differently. A visual comparison of a single grid square in the resulting equivalent gridpost terrain models is presented in Figure 3. Examples of the equivalent terrain models are also presented as each algorithm is discussed. A discussion of the fifth LOS algorithm, the intersection of line segments, is also included.

DDA with Rounding

The DDA with rounding algorithm uses the gridpost terrain file. It is based on a technique called the Digital Differential Analyzer (DDA), which is used in the numerical solution of differential equations and, in graphics, to determine which pixels to select

when drawing a line on a video screen.⁶ All of the calculations are done in the grid axis system where the post-to-post spacing defines one unit. The endpoints of the LOS vector are first transformed into the grid axis system. The DDA technique is then used to traverse the LOS vector from the eyepoint to the target with a constant step size, as shown in Figure 4. Each step is taken by adding increments to the X, Y, and Z coordinates of the previous step, beginning at the eyepoint. The number of steps is calculated by the equation:

$$\text{STEPS} = \text{INT}(\text{MAX}(\text{XDIST}, \text{YDIST})) + 1 \quad (1)$$

where XDIST and YDIST are the X and Y distances from the eyepoint to the target in the gridpost axis system. The ΔX , ΔY and ΔZ step sizes are calculated

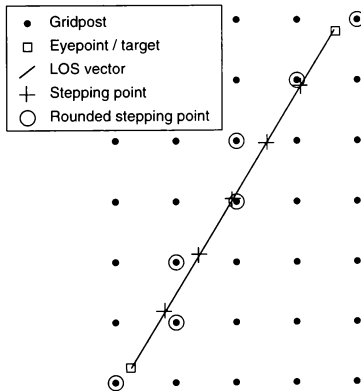


Figure 4 DDA with Rounding

by dividing the X, Y and Z distances by the number of steps. For the case presented in Figure 4 :

XDIST = 3.3
YDIST = 5.7
STEPS = 6
 $\Delta X = 0.55$
 $\Delta Y = 0.95$

The LOS vector is checked for blockage after each step is taken, with a final check at the starting position (the eyepoint). This stepping process is identical in all of the algorithms employing the DDA technique. In the DDA with rounding algorithm, the terrain height at the stepping point is defined by the terrain height at the nearest gridpost. This post is found by rounding the X and Y coordinates of the stepping point to the nearest integer. If the terrain height at this nearest gridpost is greater than the altitude of the LOS vector at the stepping point, the line-of-sight is considered blocked.

The DDA with rounding method is equivalent to a terrain model made up of square plateaus centered on each gridpost, as seen in Figure 5.

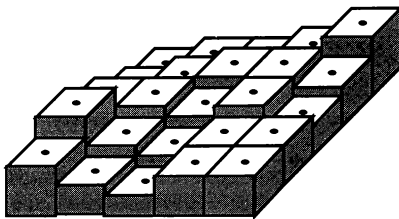


Figure 5 Rounding Equivalent Terrain Model

DDA with Three Point Interpolation

The DDA with three point interpolation method was tested to determine the increase in execution time introduced by the use of a more computationally intensive algorithm which better represents the visual database.

When drawing a visual database using a grid scheme, diagonals within each grid square can be formed by connecting the lower left gridpost to the upper right gridpost, or by connecting the upper left gridpost to the lower right gridpost. All new contoured databases at the VMS facility are built by using one of these diagonals or the other, but not

both. This suggested an approach that takes advantage of this condition.

The LOS vector is traversed using the DDA, but the terrain height at the stepping point is determined through interpolation of the three gridposts defining the upper left triangle or the lower right triangle of the surrounding grid square, depending upon which of these triangles encloses the stepping point.¹ This treatment of the gridpost terrain is equivalent to connecting posts along the lower left to upper right diagonal as well as connecting posts in rows and columns, as seen in Figure 6.

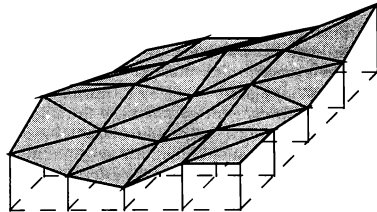


Figure 6 Three Point Interpolation Equivalent Terrain Model

DDA with Four Point Interpolation

In the four point interpolation algorithm, the LOS vector is traversed using the DDA, but the terrain height at the stepping points is calculated by performing a four way (bilinear) interpolation between the posts defining the corners of the surrounding grid square. The resulting warped plane representation can be seen in Figure 7. Unlike the three point interpolation method, four

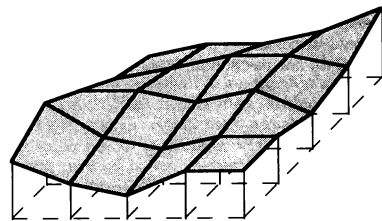


Figure 7 Four Point Interpolation Equivalent Terrain Model

point interpolation can be used when the visual database is built with arbitrary diagonals.

Grid Intersection

Another method tested, referred to in this study as grid intersection, is a two point interpolation scheme using the gridpost terrain format. It is assumed that the terrain between gridposts is defined by connecting adjacent gridposts along the rows and columns, with no diagonal connections.² A representation of this terrain can be seen in Figure 8. The LOS vector is checked for blockage whenever it crosses a row or column boundary of the terrain grid. Interpolation between the two gridposts surrounding the intersection point is used to determine the terrain height.

If line-of-sight blockage is the only information required, each axis is checked separately. Either row or column intersections may be checked first. One approach is to optimize the algorithm for speed by checking the axis with fewer intersections first. If the closest intersection to the eyepoint is required (perhaps to obtain range to intersection information), it is necessary to check row and column intersections as they are encountered when traversing the LOS vector from the eyepoint to the target. This will increase the computation time of the grid intersection method.

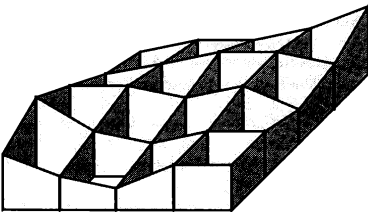


Figure 8 Grid Intersection Equivalent Terrain Model

Intersection of Line Segments

The final method tested uses the edge terrain model. Every edge on the terrain is first checked for intersections with the LOS vector projected on the horizontal plane, as seen in Figure 9a. The technique employed to determine the intersection uses the

theory of inequalities.³ A line can be represented in two dimensions by the equation:

$$Ax + By + C = 0 \quad (2)$$

where A, B, and C are constants that define the position of the line in the x-y plane. When the coordinates of a point (X_1, Y_1) on the x-y plane are substituted into this line equation, the result has the form:

$$AX_1 + BY_1 + C = D \quad (3)$$

where D is called the residual and has a unique value $D(X_1, Y_1)$. The sign of the residual D indicates on which side of the line the point falls. Points on opposite sides of a line yield residuals of opposite

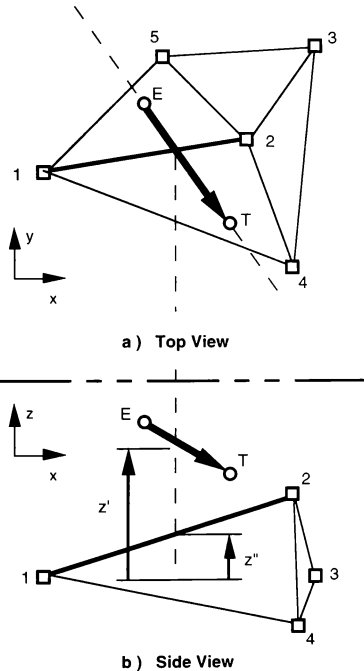


Figure 9 Intersection of Line Segments

sign when substituted into the line equation. Two line segments in the x-y plane intersect when the endpoints of each segment lie on opposite sides of the other segment.

In Figure 9a, the endpoints of edges $\overline{12}$, $\overline{14}$, and $\overline{15}$ produce residuals of opposite sign when substituted into the equation of the LOS vector, \vec{ET} , indicating these edges intersect the infinitely long line containing \vec{ET} in the x-y plane.

A second test is required to determine if the intersection lies between the endpoints of the LOS vector. In Figure 9a, substitution of the endpoints of LOS vector \vec{ET} into the equations of the candidate edges $\overline{12}$, $\overline{14}$, and $\overline{15}$ yields residuals of opposite sign only in the case of edge $\overline{12}$. This indicates that edge $\overline{12}$ intersects the LOS vector \vec{ET} in the x-y plane.

The two dimensional equations of the LOS vector and the intersecting edge are then solved simultaneously to find the coordinates of the intersection point in the x-y plane.

Interpolation is then used to determine terrain height and LOS vector height at the intersection point. If the terrain is higher than the LOS vector at the intersection point, the line-of-sight is considered blocked. In Figure 9b, the LOS vector height (z') is greater than the terrain height (z'') resulting in unblocked line-of-sight.

Test Results

The first step in testing the algorithms was to select a terrain file. For the gridpost terrain file, a 902 by 772 array was used with a 10 meter post spacing, resulting in a mapped terrain covering an area approximately 9 kilometers by 8 kilometers. The edge terrain file consisted of 650 edges and covered an area of 90 square kilometers. The average edge length was 377 meters, with a minimum of 34 meters and a maximum of 2,500 meters.

Timing

Timing of the various algorithms was accomplished by integrating the LOS software with an existing real-time simulation. The real-time computing system at the VMS facility is equipped to measure the execution time of arbitrary software segments within a real-time frame.⁵

All of the line-of-sight determination algorithms tested exhibit maximum computation time when the line-of-sight is unblocked, since all of the algorithms terminate at the first indication of a blocked line-of-sight. The eyepoint and target locations for the timing study were initialized such that the line-of-sight would always be unblocked. Each algorithm was individually timed for 2,500 iterations while recording the timing marker values for each iteration onto disk for later processing. This procedure was repeated several times with different distances between the eyepoint and target and the resulting files were then processed to determine the average execution time per iteration for each algorithm.

Figure 10 shows a comparison of the average execution time for the four algorithms which use the gridpost terrain file. Due to its simplicity, the DDA with rounding was the fastest algorithm tested. The three point and four point interpolation algorithms are also based on the DDA, but require more computation for each point checked. With a 10 meter post spacing and a range to target of approximately 500 meters, the interpolation algorithms take twice as long as the rounding algorithm. (0.2 versus 0.1 milliseconds) When the distance is increased to 5.0 kilometers, the interpolation algorithms take 4 times as long as the rounding algorithm. (1.3 versus 0.3 milliseconds) Execution time of the grid intersection algorithm falls between the execution times of the DDA with rounding and the DDA with interpolation algorithms. At 500 meters, the grid intersection algorithm takes one and a half times as long as the rounding algorithm. (0.15 versus 0.1 milliseconds) At 5.0 kilometers, it takes almost three times as long as the rounding algorithm. (0.8 milliseconds versus 0.3 milliseconds)

For each of the gridpost terrain based algorithms, the parameter controlling execution time is the number of steps checked for blockage along the LOS vector. The number of steps is a function of the distance to the target and the resolution of the grid, both of which determine the number of rows and columns crossed by the LOS vector. As a result, the execution times in Figure 10 are valid for half the distances if the grid resolution is doubled.

For the intersection of line segments algorithm, the execution time is independent of distance since the algorithm searches through all the edges without regard to eyepoint-to-target distance. The parameter controlling execution time is the total number of edges. The timing test for this algorithm consisted of varying the number of edges. The line segment algorithm timing measurements are shown in

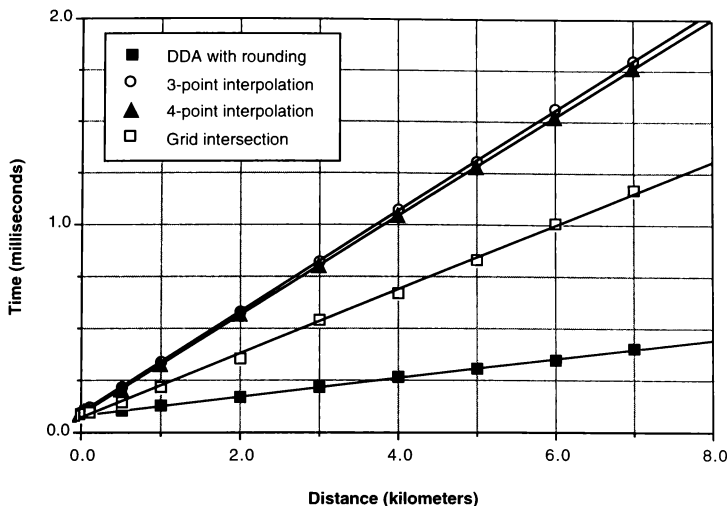


Figure 10 Timing Comparison for 10 Meter Post Spacing

Figure 11. As with the algorithms using the gridpost terrain, the maximum execution time occurs when the line-of-sight is unblocked, since the algorithm terminates at the first intersection.

kilometer in 0.36 milliseconds. The grid intersection algorithm can check up to 1.8 kilometers and the DDA with rounding algorithm can check up to 6.0 kilometers. For this non-cluttered, low resolution

The time to compute the line-of-sight for the two terrain types is a function of two independent variables : 1) the number of steps checked along the LOS vector, and 2) the number of edges in the database. A direct comparison is not possible, but an effort will be made to show a timing comparison for the two types of terrain.

The 90 square kilometer edge terrain file used in the timing tests represents a typical visual database at the VMS facility that is not built on a uniform grid. For comparison, consider a 50 square kilometer edge terrain file, or 7.07 kilometers in width and length, created assuming the same number of edges per square kilometer as the test file. This would have 362 edges and, based on Figure 11, would result in a maximum computation time of approximately 0.36 milliseconds. This time can be used with Figure 10 to determine the maximum length of an LOS vector in the algorithms using a gridpost terrain file with 10 meter resolution. The two DDA with interpolation algorithms can only check LOS distances under 1.0

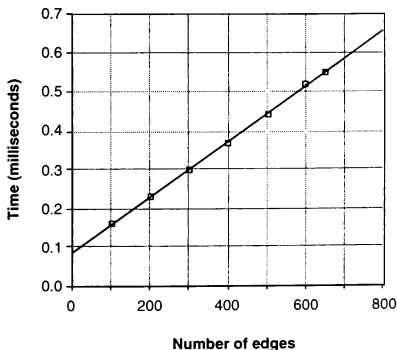


Figure 11 Line Segment Algorithm Timing

example, the line segment algorithm will check LOS distances up to 10 kilometers in 0.36 milliseconds. In the same time, the best the algorithms using the 10 meter gridpost spaced terrain can do is 6 kilometers with the rounding algorithm.

A typical visual database at the VMS facility that is built on a uniform grid uses a post spacing in the range of 100 to 250 meters. If an edge terrain file were created from such a visual database covering 50 square kilometers with a post spacing of 250 meters, it would contain approximately 2,500 edges. Extrapolating the line in Figure 11 results in a computation time of approximately 1.9 milliseconds. As the number of edges, and thus the resolution, in a given area is increased, the gridpost algorithms perform better in comparison to the intersection of line segments algorithm.

Another factor, not included in the timing study, which contributes to a larger number of edges is the presence of obstacles, such as trees and buildings. As discussed in the terrain comparison section of this paper, when using the gridpost terrain representation, the tree and building heights may be added to the terrain height at appropriate gridposts without increasing the required number of posts. When using the edge terrain representation, however, each obstacle adds to the number of edges, thus further increasing the computation time of the line segments method.

Line-of-Sight Comparison

The line-of-sight comparison portion of the testing involved driving the eyepoint and target through predetermined paths, which resulted in instances of both blocked and unblocked line-of-sight. The tests lasted anywhere between one and five minutes with all of the algorithms running sequentially in the same frame. Line-of-sight information for each algorithm was recorded once every frame (every 25 milliseconds).

The three test scenarios included a stationary eyepoint with the target flying a circular path, an eyepoint flying a circular path about a stationary target, and an eyepoint and target both flying non-intersecting circular paths. Eyepoint to target distances were anywhere from 350 meters to 3,000 meters.

The results of all gridpost terrain based algorithms were similar. Three typical 100 frame samples are presented in Figure 12. A solid line shows unblocked line-of-sight whereas spaces show

blocked line-of-sight. Samples 1 and 2 are examples of the maximum scatter found between the algorithms, whereas Sample 3 is an example of the typical scatter. The detection times between blocked and unblocked line-of sight were only a few 25 ms frames apart when comparing the DDA with rounding algorithm to the interpolation algorithms, but these differences generally disappeared in less than 200 milliseconds when eyepoint and target were moving. This time is small when compared to weapon acquisition time and pilot response time. The DDA with rounding and the grid intersection algorithms showed very good agreement.

Each of the gridpost terrain based methods can result in slight errors in the line-of-sight. The stepping points generated by the DDA based algorithms fall somewhere within each grid square. But, that point may not represent the highest terrain in the grid square along the LOS vector, thus

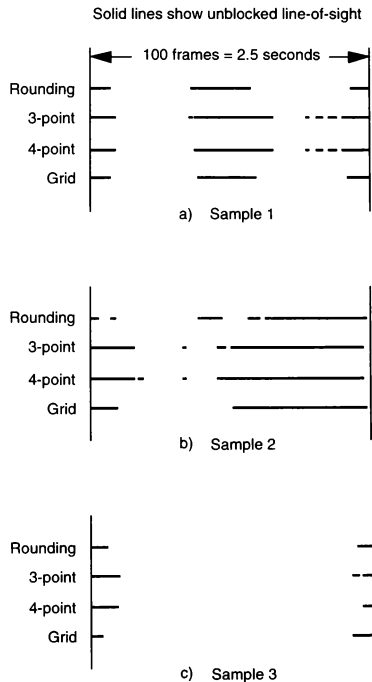


Figure 12 Line-of-Sight Comparison

resulting in slight line-of-sight errors. Also, if the LOS vector is perpendicular to a diagonal ridge line, the grid intersection method could check for blocked line-of-sight along the lowest edges in the grid square, which could result in line-of-sight errors. The magnitude of these errors is directly proportional to the post spacing used in the gridpost terrain file.

The edge terrain is an exact representation of the planes in the visual database and the use of the line equations generated by the vertices gives exact terrain height at the boundaries of the planes. This makes the line segment algorithm with the edge terrain the most accurate method. Unfortunately, its computation time is excessive for large complex databases, such as those resulting from the use of DTED terrain files. Any attempts to decrease the computation time by decreasing the number of edges also decreases how faithfully the stored terrain represents the original terrain.

Algorithm Selection

Based on the speed advantage, the DDA with rounding was selected as the baseline line-of-sight determination algorithm. Another contributing factor is the use of the gridpost terrain file which exhibits short turnaround times when modified and is easily generated from DTED files.

The algorithm is also tunable to a particular project because the terrain file storage requirements and execution time can be traded for accuracy. Since the number of stepping points is based on the number of crossings of grid rows and columns, the execution time is directly proportional to both the eyepoint to target distance and the resolution of the terrain file. For instance, if it is acceptable to double the required execution time for a given terrain resolution, the gridpost spacing may be cut in half, resulting in twice the resolution at approximately twice the execution time. The efficiency of the rounding algorithm allows for an increased grid resolution while maintaining a speed advantage. The timing results show that the rounding algorithm would still be the fastest tested if its terrain resolution were doubled.

Simple programming modifications can provide even greater speed advantages. One such possible modification is a range based step size. When using a fine mesh size, the step size can be doubled when the distance from the eyepoint to the target exceeds a given range, thus treating the terrain file as if it

had lower resolution when the distances are large. At smaller distances, the full resolution is used.

Execution time may be reduced by storing the location of LOS vector blockage from the previous frame. If the change in relative positions of the eyepoint and target between frames is small, the algorithms may be started at or near this previously blocked point.⁴ Computation time will be reduced if the LOS vector remains blocked by the same obstruction. Unfortunately, if a new obstruction blocks the LOS vector at a location closer to the eyepoint than the previous point of obstruction, the correct range to first blockage information may not be detected.

It may seem that even the more computationally intensive algorithms are extremely fast. However, when several line-of-sight determinations are required in a small frame time, the cumulative effect can result in significant time requirements, as will be seen in the forward-looking sensor example described next.

Forward-looking Sensor Example

One example of using the DDA with rounding algorithm is in a forward looking, scanning array sensor as implemented at the VMS facility.⁷ The specifications of the sensor were as follows: 1) a 50 degree by 25 degree field of view; 2) an angular resolution of 1 degree; 3) a range of 1,500 meters; 4) a linear resolution of 10 meters; and 5) a screen refresh rate every 0.75 seconds. The approach involved creating an array of LOS vectors between the sensor and the points on a grid describing a 50 degree by 25 degree section of a sphere of 1,500 meter radius with the sensor at its center. With an angular resolution of 1 degree, the resulting task requires the processing of 1,326 LOS vectors in 0.75 seconds.

In this special case of the line-of-sight determining algorithm, the range to target was a fixed value of 1,500 meters. A grid resolution of 10 meters leads to a maximum number of 151 steps per LOS vector. The aircraft model, system overhead, and data input/output required a large portion of each 25 millisecond real-time frame, leaving approximately 10 milliseconds per frame for LOS vector processing. In order to maintain the fidelity of the pilot-in-the-loop simulation, it was not advisable to increase the 25 millisecond frame time. To reduce the computation load on each frame, the task was broken into 30 parts, processing 50 LOS vectors per frame. The DDA with rounding method was able to

complete the processing of the 50 LOS vectors in less than 5 milliseconds. Here is a case where the speed of the DDA with rounding algorithm is very advantageous.

Each LOS vector was traversed using the DDA with rounding algorithm. If terrain was encountered, the processing of that current vector was terminated and the terrain height was recorded in an aircraft centered terrain database array. Terminating at the first intersection of terrain resulted in the proper occulting of objects blocked by obstructions. For example, the sensor could not "see" the back sides of hills. This is an example requiring determination of the nearest intersection with terrain. Any algorithm which does not always provide the closest intersection point would be unacceptable for this application.

In practice, this implementation of a forward-looking sensor performed extremely well. The terrain database stored by the aircraft exhibited the necessary accuracy to aid in modeling a terrain avoidance guidance system and the sensor model met the required specifications.

Conclusions

Five methods for determining line-of-sight in a real-time simulation were tested in order to select an optimal choice for the VMS facility at the NASA Ames Research Center. The algorithms tested were as follows:

- 1) Digital Differential Analyzer with rounding;
- 2) Digital Differential Analyzer with three point interpolation;
- 3) Digital Differential Analyzer with four point interpolation;
- 4) grid intersection algorithm employing two point interpolation; and
- 5) intersection of line segments employing the theory of inequalities.

The algorithms made use of two types of terrain files: the gridpost terrain and the edge terrain. Test results led to the selection of the DDA with rounding method using the gridpost terrain file. The gridpost terrain was desirable because it is easily modified, has acceptable storage requirements for the equipment used, and is compatible with DTED terrain files. The DDA with rounding was the fastest, most efficient algorithm tested. Although it exhibits a slight accuracy degradation due to the rounding process, its speed advantage offsets this small deficiency. The selected algorithm also proved

versatile enough to perform well in the modeling of a forward-looking, scanning array sensor.

Acknowledgments

The authors of this paper would like to thank Greg Carr for conducting the literature search, and for his assistance in testing the algorithms.

Recognition should also go to Russ Sansom and Dave Darling, both of SYRE, Inc., for first implementing the DDA technique at the VMS facility.

References

- 1 Strong, Randolph P., and Winn, Michael C., *The Moving Platform Simulator II: A Networked Real-Time Visual Simulator with Distributed Processing and Line of Sight Displays*, M.S. Thesis, Naval Postgraduate School, Monterey California, June 1989.
- 2 Gindelberger, Brenton E., and Steenrod, Frederick J., *A Comparison of Digital Terrain Models and Terrain Masking Algorithms*, IEEE National Aerospace and Electronics Conference, 1985.
- 3 Lewis, Michael S., and Yeo, Dennis G., *Development of a Simulator for Helicopter Air-to-Air Combat*, AIAA Flight Simulation Technologies Conference, St. Louis, Missouri, July 22-24, 1985.
- 4 Toms, R.M., *Adaptive Line of Sight Algorithms*, Lawrence Livermore National Laboratory, Livermore, California, UCRL-53812, August 25, 1987.
- 5 *μTau Real Time System User's Guide*, NASA Ames Research Center, Moffett Field, California, prepared under Contract No. NAS 2-12859, November 1992.
- 6 Harrington, Steven, *Computer Graphics: a Programming Approach*, McGraw-Hill, Inc., 1983.
- 7 Zelenka, Richard E., Swenson, Harry N., Hardy, Gordon H., and Dearing, Munro G., *Simulation Development of a Forward Sensor-Enhanced Low-Altitude Guidance System*, submitted to the AHS National Technical Specialists Meeting, Fairfield County, CT., October 6-7 1993.

TERRAIN MODELING FOR REAL-TIME PHOTO-TEXTURE BASED VISUAL SIMULATION

Dr. Venkat Devarajan and Dr. Donald E. McArthur
Department of Electrical Engineering
The University of Texas at Arlington
Arlington, Texas 76019

Abstract

There are many applications, such as pilot training, mission rehearsal, and hardware-in-the-loop simulation, which require the generation of realistic images of terrain and man-made objects in real-time. One well-known approach to meeting this requirement is to drape photo-texture over a planar polygon model of the terrain. The real time system then computes, for each pixel of the output image, the address in a texture map based on the intersection of the line-of-sight vector with the terrain model. High quality image generation requires that the terrain be modeled with a fine mesh of polygons while hardware costs limit the number of polygons which may be displayed for each scene. The trade-off between these conflicting requirements must be made in real-time because it depends on the changing position and orientation of the pilot's eye point or simulated sensor.

The traditional approach is to develop a data base consisting of multiple levels of detail (LOD), and then selecting for display LOD's as a function of range. This approach could lead to both anomalies in the displayed scene and inefficient use of resources.

In this paper, a new approach has been developed in which the terrain is modeled with a set of polygons and organized as a graph with each node corresponding to a polygon. This graph is pruned to select the optimum set of nodes for each eye-point position. The novelty of the proposed graph pruning lies in the coherence suggested in the selection of terrain LODs and texture LODs. The error in terrain representation is expressed in terms of the error in the texture pixels. This allows different error criteria to be used for different applications.

As the point of view moves, the visibility of some nodes drops below the limit of perception and may be deleted while new nodes must be added in regions near the eye point. Our previous work has been generalized to include the case where the refinements of the polygonization are not simply nested and the graph is not a simple tree. This generalization prevents the occurrence of gaps in terrain representation.

1.0 Introduction

This paper addresses the methods, algorithms and data structures required for terrain elevation representation for an image generation system which is designed to produce an image of three-dimensional terrain with photographically derived texture for flight simulation and mission rehearsal applications. The overall approach in this class of image generation systems is to drape real-world photo-texture over a polygonal model of an irregular terrain, where the terrain surface is modeled by a set of planar patches [1-6]. This system produces in real time, a realistic video image of the terrain, cultural features, and targets which are visible to the pilot of an aircraft or to an airborne sensor. It is desirable to produce an image which is as realistic as possible within the constraints of real-time processing; thus only those polygons which are absolutely necessary for a high fidelity image should be selected for display. Unique requirements arise for flight simulation applications because the pilot's eye point moves through the gaming area during the course of the mission, and the required accuracy of the terrain model for a given region varies in time. Thus a method is required for processing the terrain elevation

data off-line into a *structure* that allows the real-time selection of a minimum number of polygons which accurately represents the terrain *as viewed from a particular eye point position*.

The approach discussed in this paper is to preprocess the terrain elevation data into a polygonal data base which represents the terrain surface with varying resolutions ranging from very coarse to the finest required for any eye point position. This collection of polygons is arranged in a hierarchical structure [7-17] and made available for real-time display processing. The real-time rendering requirements include the selection of a subset of the data base from the hierarchy as needed for the current eye point and creating an image for display based on these polygons. The scope of this paper is restricted to the problem of selection of the required subset of the polygon database.

2.0 Selection Criterion Of Display Polygons

Of all of the polygons in the data base, only a small subset is required for the generation of a particular frame of the display video for a given set of sensor/pilot parameters such as position, orientation, field of view, and display resolution. As stated earlier, the polygons are stored in a hierarchical graph structure with each node in the graph representing one polygon. For any node, the child nodes correspond to a set of polygons which model the terrain with greater accuracy than the parent node. The selection process consists of pruning this graph to obtain a set of terminal nodes which represent the terrain with the required accuracy. The terminal nodes of the pruned graph correspond to the polygons which are rendered by the real-time system.

The criteria used for this pruning operation may be based on the following considerations:

- a) accuracy of the surface model,
- b) field of view,
- c) hidden surfaces,
- d) atmospheric conditions, and

e) horizon effects.

The focus of the paper is on the accuracy of the surface model.

For a terrain for which the roughness varies as a function of position, a fine polygonization must be used for the rough terrain near the eyepoint while a coarse polygonization may be used in the smooth areas and near the horizon. It must be noted that, in general, there is no simple relation between R , the range to the eyepoint and the size of the optimal polygon, and that *eyepoint position must be considered in conjunction with the terrain roughness*.

In the following, we use a more refined definition of the error criterion developed in our previous work [18] in which the orientation of the polygon as a function of the eyepoint of the pilot/sensor is taken in to consideration. We assume a system which maps a unique texture on to each triangle in the terrain model and evaluate the error in the position of the texture on the display. An elevation error $\Delta E = |h_T - h_M|$ will induce an error B_p in the texture map as shown in fig.1.

In the plane which contains the line of sight vector, we define **A** as a vector between two points. The first point is the intersection of a vertical line which passes through the true elevation point with the model plane. The second point is the intersection of the line of sight with the model plane. We define **B** as a vector from the latter point to the true elevation point. Three unit vectors are defined: **n** is vertical, **d** is normal to the model plane, and **s** is directed along the line-of-sight. The model planes that satisfy the property

$$\mathbf{d} \cdot \mathbf{s} > 0 \quad (1)$$

are visible. An error in the terrain model will induce an error in the apparent location of the intersection of the line of sight vector with the texture map *which is also in the longitudinal direction*.

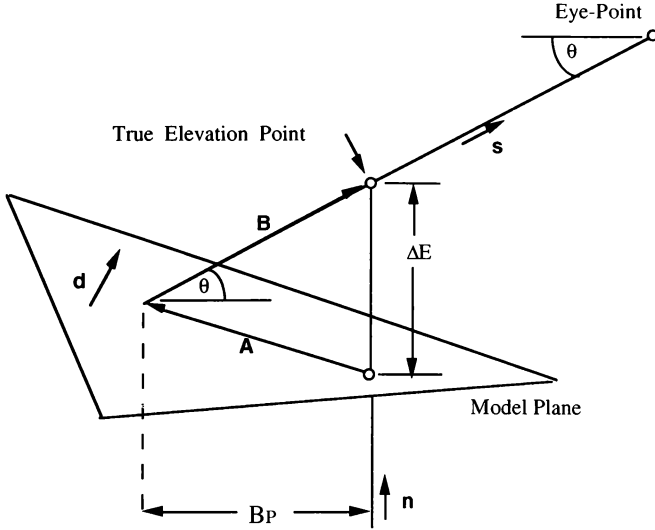


Figure 1 Geometry for calculation of B_p

Thus the hierarchical polygonal graph must be pruned in such a way that the nodes which are retained satisfy the following condition [18]:

$$[\mathbf{n} \cdot \mathbf{d}] \Delta E < \frac{d\theta (\mathbf{s} \cdot \mathbf{d}) |\mathbf{R}|}{\sin\theta \cos\theta} \quad (2)$$

This novel approach provides a coherence between the selection of terrain and texture LODs. Other published approaches have tended to separate the terrain and texture resolution selections.

3.0 Graph Pruning

An ideal terrain modeling process would produce a model with the following properties:

- i) The modeled surface is continuous for all (x, y) and for all possible pruning of the graph. This is desirable, otherwise undesirable anomalies in the displayed image may occur because of gaps in the model surface.
- ii) The model is single-valued (i.e. for each (x, y) and for all possible prunings, the model produces a unique elevation value). This is

desirable in order to avoid the problem of blending the multiple values without producing noticeable jumps in the terrain location.

4.0 Nested Refinement Of Triangular Mesh

The initial approach to this problem consisted of defining a tree structure which is related to the data base. Each triangle with its vertices and other attributes is associated with a *node* of the tree [7-9]. The root node is associated with a triangle which models a large area. Each triangle is subdivided resulting in the definition of a set of smaller triangles which cover the same area. These triangles are associated with the respective *child* nodes of the tree. The off-line data base generation task is to generate the entire graph.

The on-line, real time selection process consists of searching the tree and deleting the nodes which are not required for display. The advantage of the tree structure is that it enables the search to be organized in such a way that only the nodes which are used for display and their ancestors must be visited during the search. For 500 terminal nodes, approximately $500 \times (1 + 1/3 + 1/9 + 1/27 + \dots) = 500 \times 3/2 = 750$ nodes must be tested for each update of the display model.

A limitation of this approach is that a subdivision made by introduction of a point on the edge of a triangle may results

in a "gap" which will be visible when the model is viewed in perspective [10].

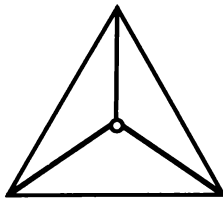
5.0 Linked Mesh

Our method was refined to avoid "gaps" by removing the requirement that the polygons be nested and allowing the triangulation to be refined by the additions of new points on the edges of triangles as shown in figure 2 [12-16]. For triangulation's derived by breaking triangles on a side, the pruning algorithm must be enhanced. In order to avoid "gaps" in the rendered image, the following rule must be imposed on the culling algorithm.

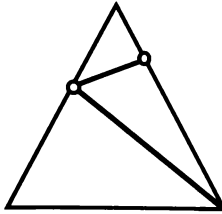
If any triangle is split on a side, then the adjacent triangle must also split [15].

To implement this rule, the data structures and algorithms must include the following features

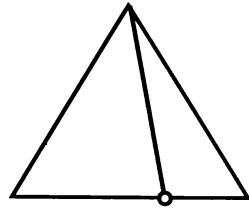
- 1) The tree structure of the data base was modified to include links defined as follows: When a triangle is split by the introduction of a new point on an edge, the adjacent triangle is also split by the same point. Each node associated with a triangle which is split on an edge contains the label of the adjacent triangle.
- 2) The graph search algorithm for generation of the display list was modified to include the requirement that if a node is retained, any linked node is also retained. This procedure results in a continuous model so that no "gaps" may occur.



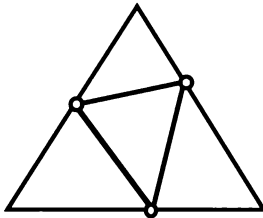
Split in Center



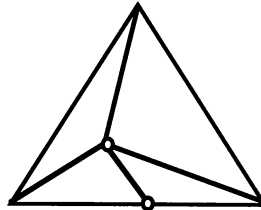
Split on two edges



Split on one edge



Split on three edges



Split in center and one edge

Figure 2. Possible refinements to a model triangulation by addition of new points [From 15].

6.0 Details Of The Tree Pruning Algorithm

This section describes the requirements for a graph pruning algorithm which is applicable to the case described in section 5. The descendants of a split node may be either rendered or further split. The nodes of the input graph may be classified as either:

- (1) S = Split
 - (2) NT = New Terminal (these nodes will be rendered), or
 - (3) D = Deleted
- according to the following rules:

- (1) The root node is S.
- (2) All descendants of S are either S or NT.

- (3) All descendants of NT are D.
- (4) All descendants of D are D.
- (5) The choice in (2) is made by the following:
 - (i) An error or visibility criterion may force a split.
 - (ii) An adjacent split forces a split.

The graph structure is assumed to contain pointers which relate any node that is split on a side to its adjacent node. These nodes will be referred to as PARED nodes. The following algorithm is proposed for node classification:

- (1) Initialize all nodes Flags to UD (undesignated)
- (2) Start at the root node and designate it as S.
- (3) Do a depth-first traverse (down to deleted nodes)

(A) If the node is not PARED, and not previously classified, test and classify according to the error criterion as S or NT.

(B) If the node is PARED and not previously classified, test both the current node and the adjacent node according to the error criterion and classify both as S or both as NT. If either is S, then both are S and all ancestors of both are S. If neither is S, then both are NT or D (NT if a child of S and D if a child of D).

(4) For each node out of the depth-first sequence which is classified as S, classify all its ancestors as S. (It should be noted here that this step may result in a reclassification to S of a node which was previously classified as NT.)

7.0 Implementation

The algorithm was coded and tested on a SUN 4/280 computer. It was then ported to an SGI system and integrated into the real-time demonstration system. Testing of the algorithm on the real time system produced a terrain model with no noticeable gaps; however, there were noticeable shifts in the terrain elevation during some of the transitions between models. The cause of these shifts is believed to be the fact that the linkages between nodes allow the formation of long chains of nodes. The algorithm demands that if one node in a chain is required for display, then all nodes in that chain must be displayed. Consequently, an attempt to tune the process to eliminate jumps resulted in a model which was either too coarse or contained too many nodes.

8.0 Recommendations For Further Work

The data base generation algorithm generates the data base graph by starting at the root node and generating refinements by splitting the triangles. At each stage, a triangle may be refined by introducing new points. These points may be interior points or edge points, and the introduction of edge points results in linked triangles. The

rules for introduction of edge points allow any number of triangles to be linked which allow long chains to be formed. These rules may be modified so that no more than two triangles may be linked into one chain. It is believed that this modification would significantly improve the performance of the system.

9.0 References

1. Biesel, Heiner, "From Source Materials to Data Bases: Higher Fidelity at Lower Cost", Presented at Interservice/Industry Training Systems Conference (I/ITSC) 6-8 Nov. 1990 pp. 163-171.
2. Clark, Charles L. and Cosman, Michael A., "Terrain Independent Feature Modeling", Presented at Interservice/Industry Training Systems Conference (I/ITSC) 6-8 Nov. 1990 pp. 7-17.
3. Cosman, Michael A. "Mission Rehearsal Modeling", Presented at the IMAGE VI Conference Scottsdale, Arizona 1992, pp. 413-425.
4. Donovan, Kenneth B., "Mission Rehearsal Database Requirements and Technologies", Presented at Interservice/Industry Training Systems Conference (I/ITSC) 6-8 Nov. 1990 pp. 157-162.
5. Ellis, John and Vellinga, Ross, "Terrain Modeling for High Performance Visual Simulation", Published by the American Institute of Aeronautics and Astrophysics, Inc. 1991.
6. Ferguson, R. L., Economy, Richard L., Kelly, William A. and Ramos, Pedro P. "Continuous Terrain Level of Detail for

- Visual Simulation", IMAGE V Conference Phoenix, Arizona, 19-22 June, 1990, Pages 145-151.
7. De Floriani, Leila, Falcidieno, Bianca, Nagy, George, and Pienovi, Caterina "A Hierarchical Structure for Surface Approximation", Computers and Graphics, 1984, Vol. 8, No. 2, pages 183-193.
8. De Floriani, Leila "A Pyramidal Data Structure for Triangle-Based Surface Description", IEEE Computer Graphics & Applications, March 1989, Pages 67-78.
9. De Floriani, L. and Puppo, E. "An On-Line Algorithm for Constrained Delaunay Triangulation" CVGIP: Graphical Models and Image Processing, Vol. 54, No. 3, July 1992, pp 290-300.
10. Latham, Roy, "An Introduction to Model Switching", Real Time Graphics Vol. 1, No. 1, July, 1992, pp 6-13.
11. Pratt, David R., Zyda, Michael J., Mackey, Randall L., and Falby, John S. "NPSNET: A Networked Vehicle Simulation with Hierarchical Data Structures", Presented at IMAGE VI Conference, Scottsdale, Arizona, 14-17 July 1992, pp 217-225.
12. Scarlatos, Lori L. "Adaptive Terrain Models for Real-Time Simulation", GACIAC PR 89-07. Proceedings of the Digital/Electronic Terrain Board Symposium, 5-6 Oct. 1989, pp. 219-232
13. Scarlatos, Lori, L. "An Automated Critical Line Detector for Digital Elevation Matrices", 1990 ACSM-ASPRS Vol. 2, Cartography, pp. 43-52.
14. Scarlatos, Lori L. "A Refined Triangulation Hierarchy for Multiple Levels of Terrain Detail" Presented at the IMAGE V Conference Phoenix, Arizona, 19-22 June 1990, pp. 115-122.
15. Scarlatos, Lori and Pavlidis, Theo., "Adaptive Hierarchical Triangulation", Auto-Carto 10, 1991 ACSM-ASPRS Annual Convention Vol. 6. Pages 234-246.
16. Scarlatos, Lori and Pavlidis, Theo., "Hierarchical Triangulation Using Cartographic Coherence" CVGIP: Graphic Models and Image Processing, Vol. 54, No. 2, March 1992, pp. 147-161.
17. Tanimoto, S and Pavlidis, T. "A Hierarchical Data Structure for Picture Processing", Computer Graphic and Image Processing Vol. 4, (1975), pp. 104-119
18. Devarajan, Venkat and McArthur, Don, "Terrain Modeling for Real Time Simulation", Proceedings of ACSM/ASPRS conference, New Orleans, 1993, Vol. 1, pp. 129-138.

A PRIMARY FLIGHT DISPLAY FOR FOUR-DIMENSIONAL GUIDANCE AND NAVIGATION
INFLUENCE OF TUNNEL SIZE AND LEVEL OF ADDITIONAL INFORMATION
ON PILOT PERFORMANCE AND CONTROL BEHAVIOUR

E. Theuissen*
Delft University of Technology
Delft, The Netherlands

Abstract

A perspective display has been developed which integrates primary flight symbology with guidance and short-term navigation information. This serves to keep the pilot in the loop in a four-dimensional ATC environment, and makes use of his ability to oversee complex problems at a glance. The display can be used for both manual and for supervisory control, and has been implemented in the moving-base flight simulator of the Delft University of Technology.

Several parameters in a perspective display can be used to influence pilot performance and control behaviour. To gain more insight in the underlying relations, an experiment has been conducted to determine the influence of tunnel size, flightpath vector and flightpath predictor configurations on pilot performance and control behaviour. It was found that a smaller tunnel yields better performance, while control behaviour can be kept at a constant level by introduction of a flightpath predictor.

Introduction

To increase airspace capacity, new strategies for Air Traffic Management (ATM) are being developed¹. These strategies require accurate long-term position and time management of the aircraft. Each aircraft is required to remain inside a so-called 'bubble-in-the-sky' which moves as a function of time. This concept was already discussed in 1950 by Jones², who developed a pictorial display for aircraft navigation in which the bubbles were represented by boxes on a CRT.

In a future Air Traffic Management (ATM) environment, a Man-Machine Interface (MMI) will be necessary which allows the pilot to plan, negotiate, and execute a four-dimensional flightplan (four-dimensional navigation and guidance). It is important that the displays used to present the necessary information require minimal mental workload, are intuitively understandable, improve the pilots' situational awareness and are compatible among the different tasks which have to be performed.

In June 1990, DELPHINS (DELt Program for Hybridized Instrumentation and Navigation Systems) was initiated at the Delft University of Technology. The main goal of this program is to develop advanced presentation methods for four-dimensional navigation and guidance information, which are suitable during manual and supervisory control.

To reduce the pilots' mental workload during the navigation task and increase his situational awareness, a display has

been developed which presents three-dimensional position information and trajectory preview in an integrated fashion by means of perspective projection.

Perspective displays have been around for a while. In 1960, LaRussa discussed a multi-purpose wide field three-dimensional head-up display. This display presented a three-dimensional road in the sky. In 1968 Wilckens¹⁰ evaluated a so-called 'channel display', and found that it is possible to fly accurate approaches using the information from such a display. Nowadays, it is possible to combine the information from a perspective display with the information from an Enhanced Vision System (EVS). This can be used to provide the pilot with the opportunity to check the information presented to him under all weather conditions, day and night. Such a system is likely to increase the confidence of the pilot as it combines raw data similar to what he is used to in VFR flight with the processed data which contains the necessary cues to accomplish the four-dimensional guidance task. Furthermore, in case of an undetected malfunction of the positioning system (integrity failure), the pilot has the ability to detect it due to the independent information presented by the EVS.

Several parameters play an important role in a perspective display. McGreevy² illustrated the need to consider the effects of perspective geometry when designing spatial instruments, and developed a model to predict average performance for different values of the observer field of view and the geometric field of view. Grunwald² and Wickens⁹ examined the effects of frame of reference in a 3D display. Filarsky¹, Grunwald², and Wickens⁹ investigated the effects of integrated velocity information. Grunwald examined the effects of the level of complexity of predictive information in a perspective display. He also evaluated different predictor laws for pictorial displays³. Grunwald and Wickens have demonstrated that when using a perspective flightpath display, tracking performance can be improved by presenting the pilot with predictive position and attitude information. Wickens¹¹ studied the relations between tunnel dimension, tracking accuracy and control behaviour. The displays he used however did not present any additional information such as a flightpath vector or a flightpath predictor.

The DELPHINS Tunnel-in-the-Sky Display

The DELPHINS Tunnel-in-the-Sky display⁸ combines basic aircraft information with guidance and short-term navigation information. The idea behind this concept is that by

*PhD student, Electrical Engineering

combining guidance and navigation information, the pilot is continuously aware of where he is and where he should be, without having to integrate information from the Primary Flight Display and the Navigation Display.

By reducing the necessity to integrate information from different sources, and presenting it in a way which requires minimal interpretation (reduced semantic distance), it is expected that workload is reduced. Furthermore, because the pilot uses this information for the guidance task, which in fact is a continuous tracking task, he automatically maintains his situational awareness which benefits safety. The trajectory preview capability allows him to anticipate upcoming changes in the trajectory, while the position preview capability allows him to monitor the future system state. As the display presents status information, the concept can be used both during supervisory and manual control of the aircraft. Due to the compatibility of the interface among the two levels of control, no problems should be encountered when switching from supervisory to manual control.

With the Tunnel-in-the-Sky display, the pilot is required to fly through a synthetic tunnel which is a representation of his desired three-dimensional flight-path. Altitude, airspeed, and bank are displayed by means of separate indicators (Figure 1).

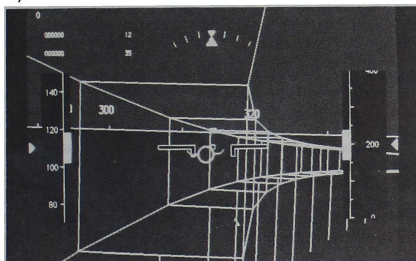


Figure 1 DELPHINS Tunnel-in-the-Sky display

The moving horizon presents attitude, while heading information is presented on the horizon line. To avoid distortions between the perspective presentation of the three-dimensional flightpath and the attitude presentation, the visible pitch attitude range corresponds to the geometric vertical field of view. To accommodate the fourth dimension, reference speed is presented by means of a bug on the speed-tape. The display also provides the possibility to present integrated speed information by means of a moving window in the tunnel.

Additional information can be presented to aid the pilot with the guidance task. This information has been divided into three levels: Unprocessed status information, processed status information, and command information.

Unprocessed status information

To aid the pilot in maintaining the correct flightpath angle, a flightpath vector can be presented (Figure 2).

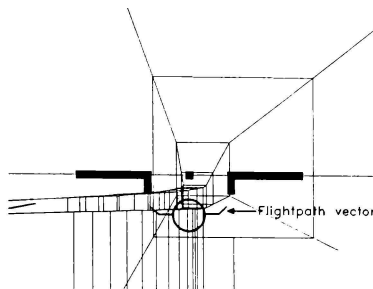


Figure 2 Flightpath vector configuration

This vector indicates the current direction of the velocity vector of the aircraft relative to the aircraft attitude symbol. Because a flightpath vector presents raw data, it is classified as unprocessed status information.

Processed status information

An airplane is a higher order dynamic system, and the pilot has to determine his control actions by predicting the future system state as a result of his actions. To aid the pilot, a predictor symbol indicating the future position and bank angle can be displayed. To resolve the position ambiguity of the predictor symbol, the cross-section of the tunnel at the position of the predictor is indicated by a transparent window (Figure 3). A predictor is classified as processed status information.

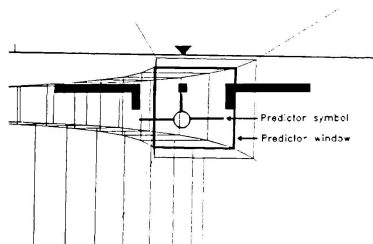


Figure 3 Flightpath predictor

Command information

Instead of presenting status information to aid the pilot controlling the aircraft, command information can be presented, e.g. by means of a flight-director.

Display parameters

As indicated previously, several parameters in the display can be used to influence pilot performance and control behaviour. The dimensions of the trajectory presented to the pilot can be used to vary the error gain and resolution with which a position error is presented to the pilot. Furthermore, the tunnel size affects the magnitude of the velocity cues which are perceived when flying through the tunnel. This will be referred to as velocity gain.

The error resolution of an element is defined as the smallest discriminable difference in position error which can be conveyed by this element.

Equation 1 gives an expression for error resolution.

$$R_{en} = \frac{2d \tan\left(\frac{fov}{2}\right)}{\text{pixels}} \quad (1)$$

In Equation 1, R_{en} represents the error resolution which is expressed as units of length per pixel. d is the distance from the viewpoint to the plane perpendicular to the viewing vector in which the element lies, fov the geometric field of view which is used for the perspective projection, and pixels is the resolution of the display expressed in pixels (Figure 4).

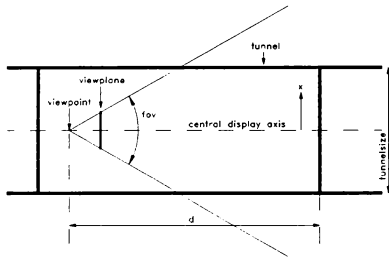


Figure 4 Display parameters

The gain of an element is defined as the displacement of this element on the display divided by the relative displacement of the viewpoint and the element in the real world. Equation 2 presents the error gain for a perspective display.

$$G_{en} = \frac{\text{screensize}}{2d \tan\left(\frac{fov}{2}\right)} \quad (2)$$

In Equation 2, G_{en} represents the error gain, and screensize the size of the screen. The parameters d and fov are the same as those used in Equation 1.

From this equation it follows that the error gain varies between 0 and a maximum value. This value is determined by distance to the points of the tunnel at the edges of the screen. These points are located closest to the observer, hence they have the biggest error gain. Equation 3 presents the relation between tunnel size and the distance to the points at the edge of the display.

$$d = \frac{\text{tunnelsize}}{2 \cdot \tan\left(\frac{fov}{2}\right)} \quad (3)$$

Equation 4 presents the general relation between the error gain and the tunnel size at a certain position on the display for a certain field of view. $K(x)$ has been introduced to indicate that among conditions where the tunnel size is changes and the field of view is kept constant, the gain also is related to the distance from the central display axis. However, $K(x)$ will remain the same when the tunnelsize is varied. Therefore, any differences resulting from a different error gain can be explained from the effect of tunnel size.

$$G_{en} = \frac{K(x)}{\text{tunnelsize}} \quad (4)$$

As can be seen from Equation (1),(2) and (3), the field of view also influences the error gain and resolution. The selection of an appropriate field of view is determined by three other criteria, the required pitch attitude range, the maximum allowable perspective distortion, and the observer field of view. Perspective distortion results in an apparent magnification of the size of an object when the viewpoint is rotated so that the object moves from the center of the display to the edge, while the viewing distance to the object remains the same. Perspective distortion is defined as the ratio of the apparent size of an object at the edge of the screen, divided by the size of the object at the center of the screen. Equation 5 presents the expression for the perspective distortion.

$$D_{persp} = \frac{1}{\cos\left(\frac{fov}{2}\right)} \quad (5)$$

The observer field of view is the visual angle of a scene as measured from the observers eye-point. If the observer field of view is smaller than the geometric field of view, the information is compressed.

The required pitch attitude dictates a certain minimum field of view, while the perspective distortion dictates a maximum field of view.

As mentioned previously, the tunnel size also affects the velocity gain. The velocity gain is defined as the velocity of an element on the display divided by the relative velocity between the observer and the element. Equation 6 presents the velocity gain.

$$G_v = \frac{x \cdot \text{screensize}}{2d^2 \tan\left(\frac{\text{fov}}{2}\right)} \quad (6)$$

The parameters screensize, d, and fov are the same as in Equation 2. The parameter x indicates the distance between the central display axis and the element. Because velocity cueing is obtained through the movement of the tunnel frames, x can be substituted by tunnelsize/2. Substituting Equation 3 into Equation 6 yields:

$$G_v = \frac{\text{screensize} \cdot \tan\left(\frac{\text{fov}}{2}\right)}{\text{tunnelsize}} \quad (7)$$

From this equation it follows that a decrease in tunnel size results in an increase in velocity gain.

Because a reduction in tunnel size corresponds to an increase in error gain and resolution, it is anticipated that tracking accuracy will increase with decreasing tunnel size until the gain is so high the system becomes unstable. This condition will dictate some minimal tunnel size.

To gain insight into the relations between error gain, level of additional information and pilot performance and control behaviour, an experiment has been conducted. Previous studies into human control behaviour show a strong relation between tracking accuracy and the presence of motion cues to the vestibular system (Hosman⁴). Therefore, the experiment was carried out in the moving-base flight simulator at the faculty of Aerospace Engineering of the Delft University of Technology.

Experiment

Subjects

Five pilots, of whom two student pilots, and one none-pilot participated in the experiment.

Experimental design

The study consisted of a 3x2 within subject design. Pilots flew three different tunnels (22.5 m, 45 m, 90 m width), in two different configurations (flight-path vector, position prediction). Each condition was replicated five times. This resulted in a total of 30 flights for each pilot. The order in which the tunnels were presented in a certain configuration was balanced to be able to compensate for possible learning effects.

Task

Pilots started their flight at an altitude of 1200 ft about 4 miles away from the runway threshold (Figure 5).

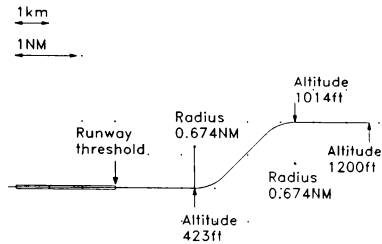


Figure 5 Plan view of the approach

The task of the pilot was to fly the curved approach as accurate as possible using the Tunnel-in-the-Sky display, and land the aircraft. Pilots were required to maintain an airspeed of 120 knots. The airspeed was indicated by a green bug on the speed-tape. No additional speed cues were presented in the display. At the beginning of the flight, the aircraft was already in the landing configuration, so no configuration changes had to be made by the pilot.

Training

Before the experiment started, pilots were briefed on the display and the approach. After the briefing, the training sessions started. To reduce the learning effect, pilots performed eight flights in each display configuration. The standard deviation of their horizontal and vertical path error was calculated for these flights and used as a measure of performance. If performance still appeared to improve after the first eight training flights, more training flights were issued.

Simulation equipment

The experiment was carried out in the three degrees of freedom moving-base flight-simulator at the faculty of Aerospace Engineering at the Delft University of Technology. The aircraft simulated was a Cessna Citation I. The instrument panel in front of the right seat is equipped with a 14" high resolution color display (Figure 6).

The viewing distance to the screen was approximately 70 cm. The screen width was 28.5 cm, and the screen height 22 cm. This resulted in an observer field of view of approximately 23 degrees horizontally and 17 degrees vertically. The Geometric field of view which was used for the perspective projection of the 3D flightpath was 52.3 degrees horizontally and 40 degrees vertically. Hence, the angular compression factor was approximately 2.4.

The reference bugs for actual airspeed and altitude are positioned on the central axis of the display (Fig. 7).

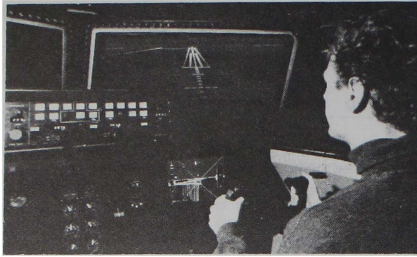


Figure 6 Flight simulator at Delft University

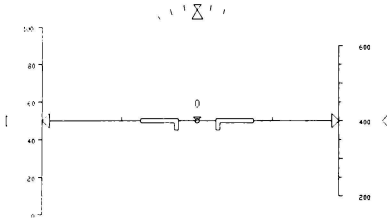


Figure 7 Position of the reference bugs

Analysis of the data

For each condition, the mean and standard deviation of the cross-track and the vertical error, and the RMS of the aileron and the elevator deflections were computed. Figure 8 presents an overview of pilot performance for the six different conditions, while Figure 9 shows pilot control behaviour. In both figures, the letter 'F' represents a flightpath-vector configuration, and the letter 'P' a flightpath predictor configuration.

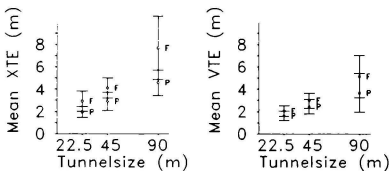


Figure 8 Pilot performance

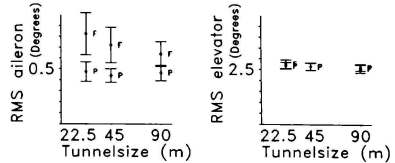


Figure 9 Pilot control behaviour

Results

Results confirm that both lateral and vertical tracking accuracy improve as the tunnel size decreases. This was found both for the flight-path vector and for the flightpath predictor configuration. Pilot performance in the flightpath predictor configuration was better than in the flightpath vector configuration.

Aileron control activity increased with decreasing tunnel size in the flightpath vector configuration, while remaining the same in the flightpath predictor configuration.

Elevator control activity remained the same for all tunnel sizes, both in flightpath vector and flightpath predictor configuration.

Aileron control activity is smaller in the flightpath predictor configuration than in the flightpath vector configuration.

A tunnel size which produces best tracking performance was not yet found, and therefore a second experiment has been conducted in which pilots were required to fly even smaller tunnels.

First results of this second experiment suggest that in the flightpath predictor configuration, the error gain can be increased beyond the limits found in the flightpath vector configuration.

Pilots generally preferred the flightpath predictor configuration. During the training flights some pilots perceived misleading cues from the flightpath vector which resulted in shortcutting the curves. It appears that this is due to mistaking angular direction information for spatial position information. Because of the dimensional compatibility of the predictor symbol and the tunnel presentation, this problem does not occur in the flightpath predictor configuration.

Discussion of the results

In the flightpath vector configuration, the results are comparable to those described by Wilckens¹¹. The fact that Wilckens found a larger tunnel size for optimal lateral tracking performance may have several causes, e.g the absence of motion cues in his experiment, the use of a more difficult to control aircraft model or a more display specific related cause.

The fact that accuracy in the vertical direction still improves with tunnels smaller than the optimal size for the lateral

control task is confirmed by the results of our second experiment.

In the flightpath vector configuration, the relation between aileron and elevator control activity and the tunnel size show the same behaviour as found by Wilckens, who did not use a flightpath vector.

As mentioned before, control activity is not significantly influenced by the tunnel size when a flightpath predictor is added. This can be explained as follows: When flying the tunnel display without predictor, feedback on position error is obtained only by the image of the tunnel. As explained previously, the error gain and resolution increase with decreasing tunnel size. The increase in error gain results in an increase in control activity and an increase in accuracy.

When flying the flightpath predictor configuration, the error gain presented by the predictor symbol is independent of the tunnel size. The fact that control activity did not change among the various tunnel sizes, indicates that the pilots use the information presented by the predictor instead of the raw information as presented by the tunnel. A smaller tunnel size does however yield a smaller prediction window. As a result, pilots should be able to fly more accurate because it is easier to determine the center of the prediction window. This is confirmed by the results.

Future research

In the current experiment, pilots were required to maintain an airspeed of 120 knots. In a future four-dimensional ATC scenario, it is anticipated that both position and velocity control must be used to correct time errors. Therefore, it is important to display information about all four dimensions to the pilot. Integration of the fourth dimension (time) into the display can be accomplished in a number of ways. First of all, time or time error can be presented numerically to the pilot. Because the pilot controls aircraft position and velocity, he has to translate the time error into a position/velocity error. Therefore, it may be more effective to display the time error as a velocity or a position error. In case the time error is converted into a position error, the information can be integrated in the perspective display. Grunwald² and Wickens⁹ claim an advantage in the integrated object representation of airspeed, while Filarsky¹ found that most pilots had trouble maintaining the correct airspeed. These differences may be contributed to the fact that Wickens integrated velocity information in his predictor symbol which presents future status information, while Filarsky used command information by means of a lead-plane, which can result in an oscillatory behaviour. Wickens, however does not report comparing the integrated velocity presentation with a conventional airspeed indicator. Furthermore, Grunwald compared velocity presentation by means of a digital readout with the integrated velocity presentation, which he found to be superior. He reports large overshoots in velocity error and throttle displacement which he contributes to the lack of damping cues for velocity control with the digital readout. The problem with a digital readout is that it is hardly possible to estimate the rate of change, but when velocity is presented by

means of a moving tape or a conventional dial instrument, rate of change can be estimated, which may provide the pilot with the necessary damping cues.

During the next experiment, the influence of fully integrated time information on pilot performance will be compared against the reference speed indication. In this experiment the pilot will either receive speed command information by means of a reference bug on the speed tape which is controlled by the 4D guidance algorithms, or by means of position information. In the latter case, a target window is presented in the tunnel which indicates his desired position in five seconds. By matching the window which indicates his predicted position in five seconds with the window indicating his desired position at that time, the pilot is able to control both velocity and position to null the time-error.

Conclusion and recommendations

The size of the perspective flightpath can be used to help the pilot obtain a certain required performance. Introduction of the flightpath predictor helps increasing the pilots' performance, while reducing control activity. Such a display combines the best properties of two concepts. It presents raw positional and attitude information which contributes to the pilots situational awareness and increases his level of confidence, and it presents processed status information which allows him to spend less effort to control the system.

Combining angular driven symbology, e.g. a flightpath vector, with a perspective display can lead to interpretation problems. Hence, special attention must be paid to the dimensional compatibility of the various stimuli which are presented to the pilot on a perspective display. This may also be important when data from an Enhanced Vision System (EVS) is combined with angular driven symbology on head-up or head-down displays.

References

- [1] Filarsky, S.M. and Hoover, S.W. (1983). The command flight path display. Naval Air Development Center, Warminster, PA.
- [2] Grunwald, A.J. (1984). Tunnel Display for Four-Dimensional Fixed-Wing Aircraft Approaches. J. Guidance, vol.7, No.3, May-June 1984, pp.369-377.
- [3] Grunwald, A.J. (1985). Predictor Laws for Pictorial Flight Displays. J. Guidance, Vol.8, No.5, Sept.-Oct. 1985, pp.545-377.
- [4] Hosman, R.J.A.W., van der Vaart, J.C. (1981). Effects of visual and vestibular motion perception on control task performance. First European Annual Conference on Human Decision Making and Manual Control. Delft University of Technology.

- [5] Jones, L.F., Schrader, H.J., Marshall, J.N. (1950). Pictorial Display in Aircraft Navigation and Landing. Proceedings of the I.R.E., pp.391-400.
- [6] McGreevy, M.W., Ratlaff, C.R., Ellis, S.R. (1986). Virtual space and two-dimensional effects in perspective displays. Proceedings of the Twenty-First Annual Conference on Manual Control, NASA Conference Publication 2428.
- [7] PHARE medium term scenario, 2000 to 2015. Eurocontrol, Issue 1, July 1990 (Draft).
- [8] Theunissen, E. (1993). The DELPHINS Tunnel-in-the-Sky program. Delft University of Technology, Faculty of Electrical Engineering. - to be published -.
- [9] Wickens, C.D., Haskell, I., and Harte, K. (1989). Ergonomic Design for Perspective Flight-Path Displays. IEEE Control Systems Magazine, June 1989, pp.3-8.
- [10] Wilckens, V. and Schattenmann, W. (1968). Test Results with New Analog Displays for All-Weather Landings. Problems of the Cockpit Environment, AGARD CP No. 55.
- [11] Wilckens, V. (1973). Improvements in Pilot/Aircraft-Integration by Advanced Contact Analog Displays. Proceedings of the 9th Annual Conference on Manual Control, Massachusetts Institute of Technology.

A COMPARATIVE EVALUATION OF THREE TAKE-OFF
PERFORMANCE MONITOR DISPLAY TYPES

J.J.L.H. Verspav and R. Khatwa

National Aerospace Laboratory, NLR
PO Box 90502,
1006 BM, AmsterdamAbstract

This paper focuses on a pilot-in-the-loop part-task simulation trails aimed at establishing the relative merits of three alternative Take-Off Performance Monitor (TOPM) display Types when compared to take-offs conducted without the aid of a TOPM. For many years it has been realised that a TOPM could potentially enhance take-off safety. There has been considerable activity aimed at developing a reliable TOPM during recent years at NASA-Langley, Bristol University and the NLR. However a widely acceptable solution still has to emerge. Fundamental to the objective of this investigation is to establish whether a TOPM (and what Type) could actually improve pilot decision making. Qualitative (ie questionnaires) and quantitative (ie measured pilot performance) results of the simulation study are presented herein.

Results show that a TOPM is able to enhance Go/No-Go judgement in certain take-offs where a performance deficit is present. A so called Type III system, which has the ability to predict both continued take-off status and stopping performance, offers the largest potential with respect to improving take-off safety.

Abbreviations

V_R	Rotation speed	m/s
V_1	Decision speed	m/s
V_2	Take-off safety speed	m/s
LOW, MED	Magnitude of performance deficit/surplus	
HIGH		

Acronyms

EFIS	Electronic Flight Instrument System
IRIS	Integrated Raster Imaging System
ND	Navigation Display
PF	Primary Flight Display
RTO	Rejected take-off
SAE	Society of Automotive Engineers
TOPM	Take-Off Performance Monitor

Introduction

Civil aircraft safety has generally improved during the last three decades for most flight phases. Some of these improvements can be attributed to the introduction of aircraft systems such as Ground Proximity Warning System and Windshear Warning System, and more recently Traffic Alert and Collision Avoidance System (TCAS). However, the accident record indicates that take-off and initial climb safety has not improved.

Some of the take-off accidents were due to sub-standard aircraft performance and subsequent failure of the crew to recognise the abnormal situation; this prevented corrective action to be applied in a timely manner. In addition, the current decision speed (V_1) concept allows only for an engine failure and therefore does not necessarily facilitate optimum decisions under other circumstances (eg tyre failure).

A Take-Off Performance Monitor (TOPM) could potentially enhance crew GO/NO-GO judgement during take-off. Various systems have been proposed over the years, but to date none has reached operational status. Various monitoring functions and their algorithms have been defined in these studies [1-16]. However, a widely acceptable solution still needs to be established. This investigation, which is described in more detail in Ref. 17, is aimed at assessing the pilot decision making process and scrutinising the nature of the TOPM display using an IRIS workstation.

Classification of systems

The US Society of Automotive Engineers' (SAE) TOPM Aerospace Standard [18] identifies any system by TYPE and the relevant definitions are summarised below.

Type I compares achieved airplane performance to a reference performance based on all-engines. These monitors indicate any deviation from the reference performance.

Type II in addition has the ability to predict performance later in the continued take-off run.

Type III has all Type I and II capabilities and can also predict the ability of the aircraft to abort the take-off.

NLR study objectives

The NLR TOPM investigation is based on the fundamental philosophy that the pilot's most essential function is to be in command, and thus the display only informs the pilot of the take-off progress and lets him decide for himself the action to take. This study involves evaluation of the concepts and displays for each of the three monitor Types and attempts to establish whether TOPM can improve decision making and if so, which is the most effective display Type. In addition, pilot performance without a take-off monitor (ie the current situation) is investigated and thus the potential benefits of any monitor Type can be established. Both pilot and non-pilot test subjects have been employed as well as two aircraft types. The results of this study will be used to modify and optimise existing features of the most promising display Type, prior to implementation on the NLR Research Flight Simulator, for further evaluation.

Displays under consideration

The purpose of the TOPM is thus to provide an enhanced means for a pilot to monitor aircraft performance. A single display has been developed for each of the three monitor Types. For this study a traditional Primary Flight Display (PFD), in Fokker 100 format, was used as a basis for comparison, whereby airspeed data and engine/aircraft status alone were employed to make the GO/NO-GO judgement.

The Type I display appears on the PFD whereas both Type II and III information is presented on the Navigation display (NDI). The lay-out of the displays on the IRIS workstation is depicted in Fig. 1.

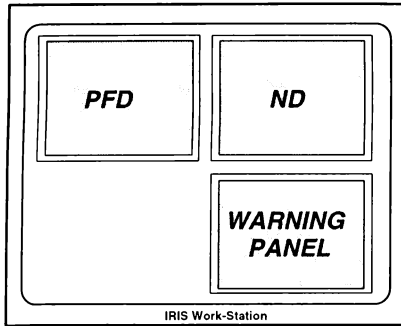


Fig. 1: Display layout

Type I display

Fig. 2 depicts the Type I display. This display is based on work reported in Ref. 8. The following are presented.

- Actual and scheduled airspeeds. The scheduled airspeed (yellow *solid* triangle) represents the airspeed which would normally be achieved at a given runway position for the reported conditions and the intended engine setting.
- Speed-trend information (magenta line) indicating a five-second prediction of airspeed. A reference speedtrend (white cursor) is based on 85% of the expected acceleration for the recommended power setting and reported conditions. Note that these cues provide a means of monitoring inertial acceleration. A 15% reduction in acceleration would give rise to an almost similar increase in distance; a 15% mandatory safety margin is incorporated in the all-engines take-off field length computations. These cues enable monitoring of a depletion in the safety margin.
- Current runway position of aircraft (yellow bar).

Note that Fig. 2 shows a Type I display for worse than expected performance.

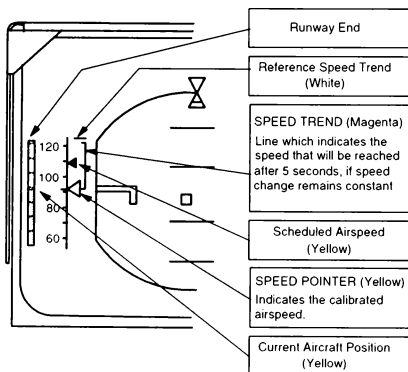


Fig. 2: Type I TOPM

Type II display

A simple form of the display presents the following information (Fig. 3).

- Current runway position of aircraft (yellow aircraft symbol).
- Predicted (yellow cross) and nominal (blue circle) runway positions which the aircraft will reach when its speed becomes V_R .
- The runway region from where a decision to continue take-off *safely* with an engine failure is possible (blue bars). 'Safely' in this context implies the ability of the aircraft to clear the departure end of the runway by at least 35 feet (screen height) at speed V_2 (take-off safety speed). However, for wet runway operations, a reduced screen of 15 feet (permitted by existing UK regulations) is employed.

Note that (a) and (b) are based on Ref. 1-5. If take-off conditions change such that the aircraft can no longer clear the fictitious screen at the runway end at speed V_2 , then the predicted V_R marker changes colour from yellow to red. This indicates a potentially dangerous continued take-off.

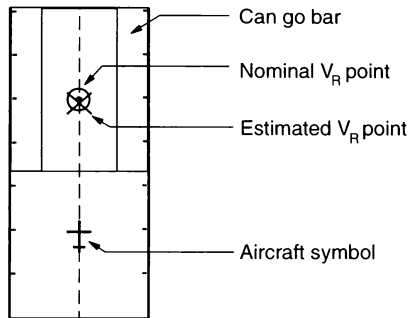


Fig. 3: Type II TOPM

Type III display

In addition to all the data presented above for the Type II display, the Type III TOPM also presents the runway region from where it would be possible to conduct a safe stop (brown bars). This display is shown in Fig. 4. The stopping computation does not allow

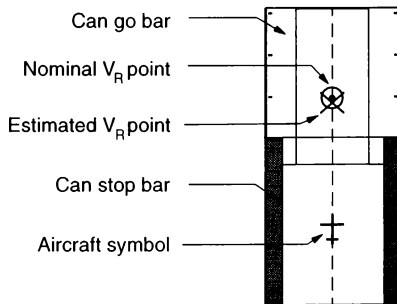


Fig. 4: Type III TOPM

credit for reverse thrust, reduced braking capacity due to burst tires is not accounted for. Note that the colours used to display the 'can go' and 'can stop' options conform to the blue (depicting sky) and brown (representing ground) respectively adopted for a conventional artificial horizon. For the Type III display evaluation, TOPM information alone can be utilised to make the take-off decisions and thus V_1 speed data was not furnished to the pilots for these tests.

Fig. 5 shows the display with five different performance levels. Note that the cross which represents the prediction of the position on the runway where V_1 is reached moves towards the departure end of the runway as performance decreases. Also note the appearance of a gap between Can-Go and Can-Stop bars as performance further decreases and eventually the disappearance of the Can-Go bars.

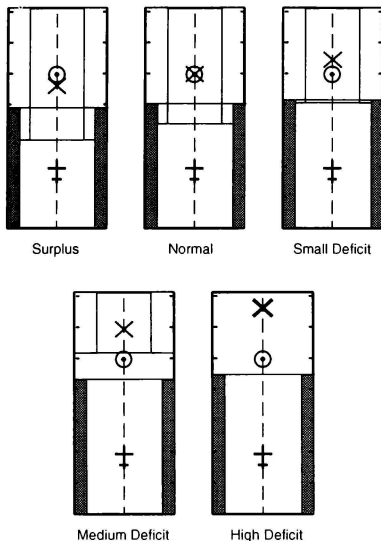


Fig. 5: A Type III TOPM with five different performance levels

TOPM Algorithm

The pre-takeoff module estimates the critical take-off lengths prior to brake-release. During ground roll these lengths are evaluated on the basis of measured data.

During ground roll the difference between expected and measured acceleration is established. This difference in acceleration is employed for Type II and III monitors to predict the continued take-off lengths (Fig. 6). Both zero- and first-order correction polynomials are available to predict the future accelerations [16]. The first-order polynomial gives the best fit if the acceleration difference varies as a function of speed, eg in the presence of slush. The algorithm establishes which of the two correction polynomials is most likely to correctly predict acceleration. Estimated wind-speed is also incorporated in the computations.

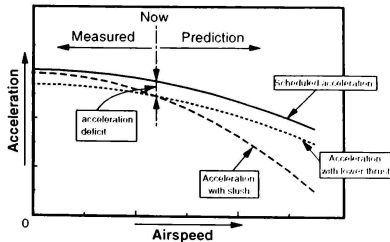


Fig. 6: Principle of TOPM predictions

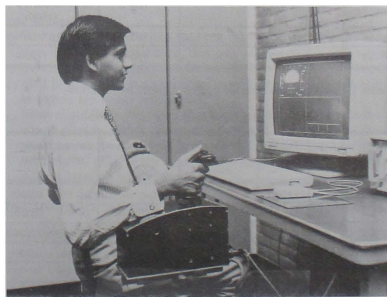
In this experiment two constant deceleration levels were incorporated in the algorithm of the Type III to predict stopping distance. These two levels represented typical mean deceleration levels for dry and wet runways.

Simulation Equipment

A Silicon Graphics IRIS 4070 GT workstation facility was employed to conduct this investigation. The workstation is interfaced with a keyboard, a control stick for pitch control (on right hand side) and a throttle box for demanding changes in thrust (left hand side). The test facility, configured for a one-man crew is illustrated in Fig. 7. This study is limited to an investigation of the pilot monitoring instruments. Three CRT type displays are presented on the workstation monitor and their arrangement is as shown in Fig. 1. The PFD and ND are presented in the form of a left-to-right arrangement respectively.

Powerplant status information was limited to an engine failure flag appearing on the warning panel.

A general warning flag, presented on the warning panel, indicates the presence of disturbances such as tyre burst, hydraulics failure, control problems and malfunctions of critical systems. Note that these events are either performance or non-performance related. For this experiment precise control of the aircraft was not under scrutiny and more importantly subject GO/NO-GO decision making was of interest. Therefore the flight controls provided were limited. The simulated aircraft was programmed to remain on the runway centre line and thus rudder and nose wheel steering were not necessary.



Housed on top of the side-stick is a notice button. Test subjects were required to actuate this button upon detecting a performance anomaly.

Both Boeing 747 and Fokker 100 aircraft models were employed in the tests. The simulation employs non-linear models for both aircraft. The aircraft used in this experiment are assumed to be equipped with an Auto Throttle System (ATS) and an Auto Brake System. For the purpose of this study both are controlled by a single throttle lever. Full take-off power is the thrust rating selected for the scenarios considered herein. However, additional thrust could be demanded by firewalling the lever to the forward limit. An RTO is executed by retarding the thrust lever to the rear limit. This initiates engine spool-down to idle and activates the Auto Brake System to achieve the maximum level of deceleration.

Test subjects

For this evaluation the 19 test subjects were divided into the three categories, defined in Table 1, namely non-pilots, twin-engined aircraft pilots, and multi-engined aircraft pilots. Pilots of twin-engined aircraft were subjected to the Fokker 100 simulation, whereas multi-engined aircraft pilots were employed on the Boeing 747 simulation. The aircraft employed for all non-pilot tests was the Fokker 100.

Subject Group	Subject experience	No.
Non-Pilots	without flying experience	7
Twin-Engine	currently flying aircraft with 2 engines	6
Multi-Engine	currently flying aircraft with 3 or 4 engines	6

Table 1 Subject groups

The pilot subjects were all qualified airline crew and most had some experience with EFIS displays. A survey of pilot flight experience is presented in Tables 2 and 3. All participants were male.

Pilot	Age	Flying Hrs.	Years of Professional flying	Present Crew position	Current Aircraft Type (Hrs)
B1	46	6000	20	Captain	B757/767 (2000)
B2	46	12000	28	Captain	A320 (1500)
B3	33	7200	11	Captain	B737-300&400 (600)
B4	34	4000	6	F/O	B737-300&400 (2500)
B5	34	3000	4	F/O	B757/767 (2000)
B6	32	3500	10	F/O	B737 (200)

Table 2 Twin-Engine aircraft group subject experience

Pilot	Age	Flying Hrs.	Years of Professional flying	Present Crew position	Current Aircraft Type (Hrs)
C1	41	11000	21	Captain	DC-10 (8000)
C2	60	18000	43	Captain	A310 (5000)
C3	32	5000	12	F/O	B747-300 (2000)
C4	45	13000	25	Captain	DC-10 (2000)
C5	65			Captain	DC-10
C6	31	4000	6	F/O	B747 400 (1000)

Table 3 Multi-Engine aircraft group subject experience

Evaluation sequence

The evaluation sequence normally was spread over two days for each test subject. Each session started with training to familiarise subjects with the workstation facility, the displays and the evaluation tasks and required approximately two and a half hours. Following the training session, a quantitative evaluation was performed for each of the four displays. A total of 40 carefully selected flight scenarios were employed for each display evaluation. These included normal, degraded and better than expected performance and malfunctions. Also various combinations of these conditions were used. Subjects were required to complete a short questionnaire, on-line, immediately after each simulation run.

Following all quantitative evaluations, participants were required to comparatively rate the displays. Subjects were also invited to provide unsolicited comments, criticisms, recommendations, format changes, etc. Finally, a verbal debriefing followed whereby the pilot was able to discuss or make any comments about any aspect of the evaluation.

Qualitative Analyses

The qualitative evaluation involves the use of questionnaires. A display rating questionnaire was completed by the test subject at the end of each test session for a particular display. This questionnaire considers aspects such as

- usage of each display element
- choice of symbology
- size of symbols
- display dynamics
- use of colour
- ease of interpretation
- amount of information provided (eg too much, not enough)
- whether display elements were beneficial or distracting
- level of mental workload
- degree of situation awareness
- overall rating for the display
- any information/format changes recommended.

Most questions are in a *tick-the-box* format and comprise five statements with opposite viewpoints at the two extreme ends of the scale; the middle rating always indicates a *neutral* position. Note that *average ratings* which enable global conclusions to be drawn are presented herein.

Ease of interpretation, usage of colour and size and choice of symbols were generally rated favourably for all displays under consideration.

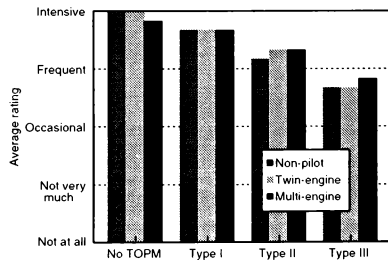


Fig. 8: Usage of actual CAS

Fig. 8 illustrates the relative usage of the CAS for all three TOPM Types, in addition to the conventional case whereby take-off is conducted without the aid of a TOPM. Average ratings for a part

cular display, for the three subjects groups, were in close agreement. As expected CAS usage for the 'no TOPM' case was intensive. Fig. 8 implies that usage decreased as the TOPM Type varied from I through to III. Average CAS usage for the Type III system was rated occasional-frequent.

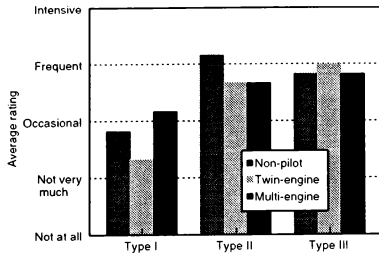


Fig. 9: Usage of current runway position

Fig. 9 illustrates the relative usage of the current runway position symbol. For the Type I system had the most infrequent use of this data. There was a marked increase in usage of the runway position symbol with the Type II system, whereas even higher ratings were attained with a Type III monitor. This trend can probably be attributed to subjects having to coordinate the aircraft symbol with the 'can go' bars associated with the Type II and III TOPMs, and in addition the 'can stop' bars for the Type II monitor.

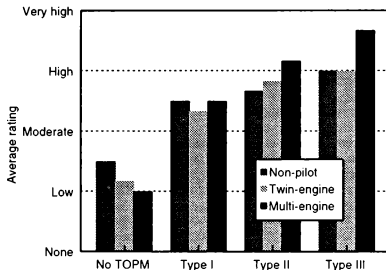


Fig. 10: Degree of situational awareness provided

Fig. 10 also illustrates a striking trend; the degree of situational awareness appeared to increase as a function of TOPM Type for all subject categories. The 'No TOPM' case provided the lowest degree of situation awareness (average rating about low) and there was a marked increase to moderate-high for the Type I display. Type II TOPM average ratings were slightly higher, whereas Type III ratings were the highest (high-very high).

The overall ratings (Fig. 11) for the displays followed a trend very similar to the situational awareness results. The average ratings for the take-offs conducted without a TOPM, for all subjects, varied between bad and fair. Type I display average ratings increased to fair-good and Type II display ratings were about good. The Type III display attained the highest average ratings (good-excellent) for all subject groups.

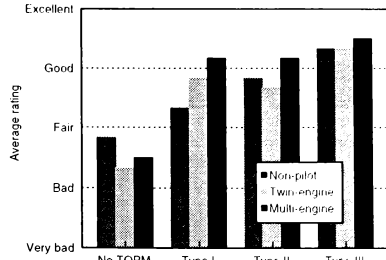


Fig. 11: Overall rating

Subject comments

Conventional take-off display:

- Several pilots commented that in this experiment they missed the physical feeling associated with aircraft acceleration and the outside view but they also expressed that even with these cues it would not be easy to gauge performance. One pilot stated "It's lousy, but at present its all we've got in the cockpit!" Another stated "... this is what we live with!" The general warning flag was considered too vague by several participants as it was non specific. This was however the intention of the general warning flag as in actual operations the source of a problem is not always obvious.

Type I display:

- The current runway position symbol generated many comments. Many subjects suggested that the information was superfluous. Several subjects indicated that this was the easiest display to scan due to it being housed on the PFD. A number of pilots stressed that the CAS cues allowed them to identify windshear conditions, whereas the speed-trend information enabled detection of deficiencies related to inertial acceleration. A significant number of non-pilot subjects considered the speed-trend cues difficult to use.

Type II display:

- Several pilots indicated that if the display indicated sub-standard performance the TOPM allowed them to restore the performance margin by addition of extra power.

Type III display:

- The appearance of a gap (see Fig. 5) indicated a critical situation. One pilot stated that "one can use the display for a total overview of aircraft performance and can recognise an impending problem much earlier than would otherwise be possible, ie trends are easily recognised." Several subjects indicated that it was easy to assimilate the whole display. Other comments reflected views such as "no changes required, almost perfect" and "a big help in determining the V₁ area. If the system proves to be reliable I hope to meet it later on future aircraft."

Other comments of a more general nature are summarised below:

TOPM was generally considered to be a useful decision making tool. For example, one subject stated "prior to the trial I was very concerned about the frequency of high speed aborts, I now see that there are occasions when TOPM allows you to continue safely." Another stated "this system (TOPM) helps a great deal in making a fast decision to abort take-off or not..." One subject said "I hope these systems

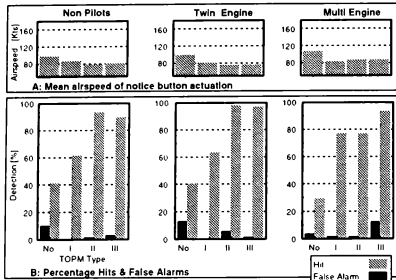


Fig. 13: All scenarios combined
Detection of performance anomalies

Detailed qualitative results

It is crucial to consider results of separate subsets of the 40 scenarios as certain effects may be obscured by presenting only the global results (ie above). The 40 scenarios have been partitioned into a large number of discrete event/performance anomaly groups and analyzed. The relevant results are discussed below.

No differences were found between the different displays and the No-TOPM case when performance was about nominal and no failures occurred, this is of particular importance as the large majority of take-offs in real life will be in this category. Small anomalies in performance did not cause the subjects to make erroneous decisions, these small anomalies will also happen in actual operations but do not pose a threat to safety.

Medium Performance Deficit, with Engine Failure

A performance deficit will cause that the point where V_L is reached will move towards the departure end of the runway. This implies that in case the take-off is continued the airplane will reach a lower height at the end of the runway than it would otherwise have reached, on the other hand in case the take-off is aborted at V_L the airplane would come to a standstill later. A Medium Performance Deficit now is defined as a performance deficit such that in case of an engine failure on a certain area on the runway it would neither be possible to continue the take-off safely nor be possible to stop within the remaining runway length. Occurrence of this situation would become apparent on a Type III with the presence of a gap between the Can-Go and Can-Stop bars (Fig. 5). The results of this subset of scenarios is examined more carefully.

The percentage of unsafe actions, as represented by the extreme colours in Fig. 14, decreased for all three subject groups as TOPM Type increased. The consequences of superior performance anomaly identification is evident in this scenario group as the medium performance deficit/engine failure combination could cause an accident without timely pilot intervention (ie. addition of extra thrust or an early abort).

The Type I TOPM has achieved the least safety benefit for the non-pilot and the twin-engine subjects, and in fact the number of unsafe continued take-offs increased for the multi-engine subjects. When using a Type I TOPM the test subjects were unable to effectively exploit the information gained from knowing that a performance anomaly was present (Fig. 13). The predictive capacity of the Type II and especially the Type III displays aided the test subjects in making better decisions. For Fokker 100 subjects Types II and III displays generated the least number of unsafe RTOs.

The three groups of subjects appear to have adopted different decision strategies. Without the aid of a TOPM, non-pilots had the tendency to continue the take-off more often than the pilot subjects. Both Fokker 100 groups used the option to apply extra thrust at low speed frequently and therefore were frequently able to continue take-offs successfully. As the multi-engine group did not apply this policy their only safe option was a low speed abort.

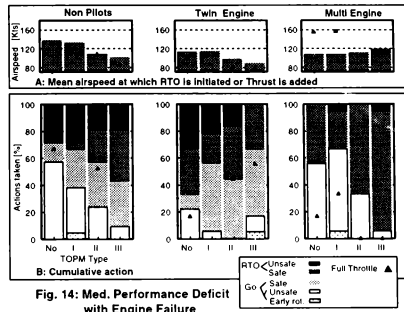


Fig. 14: Med. Performance Deficit
with Engine Failure

The potential safety benefit of TOPM in these scenarios are different for the three groups; For non-pilot and multi-engine subjects groups, Types II and III TOPM displays generated less unsafe continued take-offs, whereas for the twin-engine pilots the number of RTO overruns decreased. For both Fokker 100 groups the abort speed appeared to decrease as TOPM Type increased, whereas the converse is true for the Boeing 747 subjects (see Fig. 14a).

High Performance deficit

A performance deficit such that even without an additional failure the climb out would not be safe is qualified in this study as a High Performance Deficit. As Fig. 15 shows there is a definite decrease in the percentage of unsafe continued take-offs as TOPM Type increases. In fact there is not a single unsafe go decision for the Type III TOPM. Again there is a large difference in the decision policy adopted by the Fokker 100 and the Boeing 747 subjects.

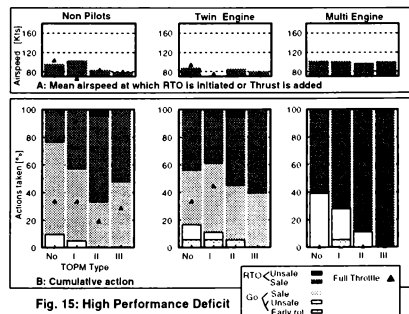


Fig. 15: High Performance Deficit

(TOPM) will be the future. I think it will be a big help for the GO/NO-GO judgement."

Several subjects stated that regardless of TOPM Type, concepts such as delegation of crew duties, high speed abort policy, etc need to be assessed for TOPM operations.

Quantitative Analyses

The five possible outcomes of each given scenario are defined as follows:

- (1) Unsafe RTO: the airplane would not have stopped before the runway end.
- (2) Safe RTO: the airplane would have stopped on the runway.
- (3) Safe continued take-off: the airplane will be able to clear the imaginary screen at the runway end with a speed of at least V_2 .
- (4) Unsafe continued take-off: the airplane would not have been able to climb over the imaginary screen at the take-off safety speed.
- (5) Early rotation: the rotation was initiated at a speed more than 5 Kts beneath V_r .

To evaluate the test subject performance during each run, the airspeed, the time after throttle advance and position on the runway associated with the following actions were recorded:

Firewalling the throttle lever to demand additional thrust.
Retarding the throttle lever to initiate an RTO.
Pulling the stick back to initiate rotation.
Actuating the notice button.

Fig. 12 illustrates the results of the outcomes for all scenarios combined. Fig. 12b illustrates the cumulative percentages as a function of display Type for each subject group. Note that the extreme colours represent the unsafe actions. The top of the light grey bar (Fig. 12b) represents the percentage of all continued take-offs (both safe and unsafe). Consider the non-pilot group, generally very few continued take-offs were initiated more than 5 Kts beneath V_{r0} , typically less than 2%. Fig. 12b shows that unsafe continued take-off decisions gradually decreased from 6% to 1% as TOPM Type increased. The number of unsafe aborted take-offs is approximately constant at about 2%. Whereas, the overall number of continued take-offs, both safe and unsafe, gradually decreases from 81% for the no TOPM case to 61% for the Type III display. Conversely the number of safe RTOs rises as the TOPM Type increases. The solid triangle in Fig. 12b indicates throttle lever application to demand maximum thrust as percentage off all scenarios. This remained constant at around 23%.

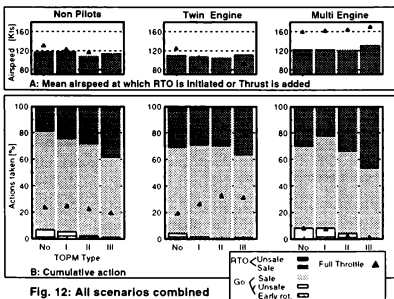


Fig. 12: All scenarios combined

Fig. 12a depicts the mean airspeed at which the abort (dark grey bar) and throttle advance (solid triangle) were initiated. For non-

pilots, and those take-offs conducted without TOPM and with a Type I display, the mean abort speed was around 120 Kts. For the Type II display the mean abort speed was somewhat lower at 108 Kts and for the Type III display the speed was 115 Kts. The mean speed at which throttle advance was initiated gradually decreased from 131 Kts for take-offs not aided by TOPM to 106 Kts for a Type III TOPM.

Results of the twin- and multi-engine airplane pilots are also presented in Fig. 12a. For the twin-engine group the number of continued take-offs conducted without a TOPM and for Type I & II TOPMs is about 70% and falls to 63% for the Type III TOPM. Unsafe continued take-offs decrease when a TOPM was used. The number of unsafe RTOs gradually decreased as TOPM Type increased and there are no unsafe RTOs with the Type III TOPM. Fig. 12a shows that the mean airspeed at which the RTOs were initiated for each display is around 110 Kts. The frequency of throttle advance during ground roll increases from 20% without TOPM to more than 30% for both Type II & III TOPMs. The mean airspeed at which this was accomplished decreases from 125 Kts to around 95 Kts.

Results of the multi-engine group indicate that for all display Types very few unsafe RTOs were conducted. Unsafe continued take-offs decreased only when Type II and III TOPMs were used; to almost zero for the Type III display. The percentage of RTOs increased as TOPM Type increased. The mean airspeed at which the RTOs were initiated is about 120 Kts for the conventional display and for the Type I and II, whereas it increased to 130 Kts for the Type III system.

Fig. 12b also shows that the multi-engine group generally used the option to advance the throttle lever during ground roll much less than the other two groups. It is evident from Fig. 12a that when this facility was used, it was done so at a much higher airspeed than the other two groups, indicating an alternative use of the maximum power option.

Detections

Test subjects were required to press the notice button when performance was considered to be non-standard. For this particular task one of four outcomes is possible, ie:

Hit	The notice button is pressed when performance is non-standard.
False Alarm	The notice button is incorrectly pressed when performance is standard.
Miss	The subject fails to press the notice button when performance is non-standard.
Correct Rejection	The notice button is not pressed whilst the take-off conditions are standard.

The latter two complement the former two outcomes and clearly the ideal score is 100% hits and 0% false alarms.

For all subject groups the percentage of hits increased as TOPM Type increases, while the number of false alarms is generally very low (Fig. 13b). Fig. 13a presents the mean airspeed at which the notice button was pressed for the hit case. For all subject groups, the mean airspeed decreased from approximately 100 Kts for the no TOPM case to around 80 Kts for the TOPM aided take-offs. For a mean take-off acceleration of about 4 Kts/s, this implies that the TOPM enabled subjects to notice the anomaly some 5 seconds earlier. Also note that both F-100 subject groups advanced the throttle lever at about the same airspeed as they noticed a performance anomaly, whereas the B-747 group did not.

The two Fokker 100 subject groups frequently used the extra power option at an airspeed of typically 80 Kts or elected to abort the take-off (at around the same airspeed). The multi-engine group did not apply extra power during ground roll and aborts were typically initiated at speeds around 100 Kts for all display cases. Note that even though the substandard performance detection capability of the Type I TOPM has much improved as compared to the No-TOPM case (Fig. 13), the number of unsafe continued take-offs is still much higher with a Type I than with a predictive system.

Failures occurring close around V_1

Within this subset the failures occur just before or just after V_1 , and performance up to occurrence of the failure is normal or has only a minor anomaly which does not significantly affect aircraft performance. When no TOPM or the Type I or II is used, the three subject groups continued the take-off in the majority of the failure scenarios (Fig. 16). However when the Type III display was used, only around half of the take-offs were continued. The aircraft was able to stop on the runway for almost all aborts made. The relatively high number of aborts with a Type III TOPM is a point of concern because the current training trends aimed at improving take-off safety promote a more go-minded attitude [19]. It should be noted that this increase in the number of aborted take-offs is not inherent to TOPMs in general, as Type I & II systems did not increase the number of aborted take-offs under these circumstances.

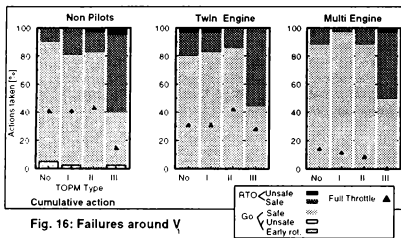


Fig. 16: Failures around V_1

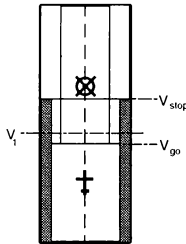


Fig. 17: Relative position of V_1 in the Type III display

The high number of RTOs may be generated as a consequence of the Type III implementation using a decision criteria based on the position of the aircraft relative to the "Can Go" and "Can Stop" bars, whereas the other three schemes utilise the V_1 concept.

No strict procedure was specified for use with the Type III and it was left to the pilot how he would use the information provided by the monitor. The transition from a stop-minded to a go-minded

attitude (effective decision point) now became located approximately in the middle of the overlap area of Can-Go and Can-Stop bars. The V_1 point, which serves as decision point for the other displays, however, in most cases is located closer to the beginning of the overlap area (see Fig. 17). Therefore the effective decision point became located further down the runway with usage of the Type III TOPM. Consequently the number of aborted take-offs was higher.

Subject reaction times

A concern raised previously about TOPM implies that as display Type increases the complexity also rises and that this could lead to an increase in pilot reaction time. The time from a failure developing to the initiation of an abort was measured to investigate this. Mean reaction times are plotted in Fig. 18. There are only minor differences in reaction times between TOPM types, with no trend towards an increasing reaction time. Also the individual reaction times of the three subject groups to engine failures (EF) and general failures (GF) are presented in Fig. 18. No distinct difference exists between the two failure types, however, an effect which can be seen is the difference in reaction times between two groups, with pilots of twin-engine aircraft being the fastest (around 1 sec.) and non-pilots the slowest (around 1.6 sec.). The difference between the pilot and non-pilot groups can be explained by the better training of pilots and their selection criteria as compared to the non-pilots. The difference between the two pilot groups is probably due to the age difference between the group of twin-engine pilots and multi-engine pilots. On average twin-engine pilots are younger (see Tables 2 and 3).

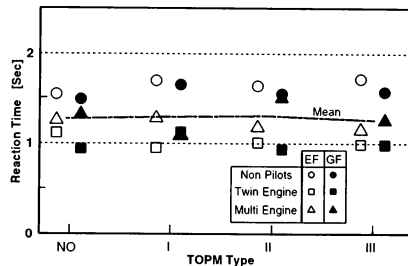


Fig. 18: Reaction Time: from Failure to Abort

Conclusions

The detection of performance anomalies increased substantially with all TOPM Types. The predictive Type II and III monitors provided the highest detection capability; almost all performance anomalies were detected, and in fact less than half were detected without a TOPM. The comparative Type I monitor detection performance is slightly less than Type II and III at around 70%. TOPM also allows detection of anomalies earlier (about 5 seconds earlier).

An increase in take-off safety does seem possible in two classes of scenarios with a TOPM. These classes are:

- in the presence of a performance deficit an engine failure develops and consequently it will no longer be possible to abort or continue safely at V_1 ,
- a performance deficit exists large enough such that safe climb out is not possible.

No safety benefit with a TOPM was found in those scenarios where an engine or general failure occurred in the absence of

a significant performance anomaly; without TOPM number of overruns was almost zero in the experiment.

The increase in safety, if any, was marginal with a Type I TOPM. The Type III system has the largest potential safety benefit. The Type II scheme performed better than the Type I system, but not as well as the Type III. So however the Type I monitor gave the subjects a large increase in the detection capability of performance anomalies, it does not improve the decision making process. Clearly the predictive capabilities of the Type II and III monitors are needed to improve pilot decision making.

- Except for the two classes of scenarios where safety did improve no side effects were found with the exception that the Type III TOPM, in its current implementation yields a higher number of aborts.
- There appears to be no significant differences in strategies adopted by non-pilots and pilots. However, pilot reaction times (time between failure and initiation of RTO) were shorter.
- The twin-engine pilots generally applied additional power when substandard performance was detected, whereas multi-engine pilots did not.

Recommendations

- Concentrate the study on a Type III TOPM, as that yields the highest potential safety benefit.
- To investigate the higher number of RTOs of a Type III TOPM and possibly correct that problem area.

Scrutinise the very important aspects of crew interaction and crew procedures.

Acknowledgements

This study was conducted under a contract awarded by the Netherlands Agency for Aerospace Programmes (NIVR). The authors also wish to thank the following for their support:

- Air Holland, British Airways Plc, KLM Royal Dutch Airlines, Lufthansa German Airlines, Martinair and their pilots for participating in the trials.
 - All non-pilots for taking part in this study.
- The NLR Flight Simulator Group.
- The NLR Human Factors Group, in particular Hilda Folkerts and Peter Jorna.

References

- [1] Srivatsan S. *Design of a Takeoff Performance Monitoring System*, DE Dissertation, University of Kansas, Department of Aerospace Engineering, 1985; also published as *NASA CR-178255*, 1987.
- [2] Srivatsan R, Downing D R, Bryant W H. Development of a Take-off Performance Monitoring System, *NASA TM-89001*, 1986.
- [3] Middleton D B, Srivatsan R. Evaluation of a Takeoff Performance Monitoring System Display, *AIAA Journal of Guidance and Control*, Vol. 12, No. 5, Sep-Oct 1989.
- [4] Middleton D B, Srivatsan R, Person L H. Simulator Evaluation of Takeoff Performance Monitoring System Displays, *AIAA Flight Simulation Technologies Conference*, Atlanta, September 1988.
- [5] Middleton D B, Srivatsan R, Person L H. Takeoff Performance Monitoring System Display Options, *AIAA Flight Simulation Technologies Conference*, Hilton Head Island, August 1992.
- [6] Grover J H H. *Handbook of Aircraft Performance*, BSP Professional Books, 1990, pp. 107-112.
- [7] Cleary P J, Kelman L S, Lloyd S, Horn L. Aircraft Performance Margin Indicator, *US Patent No. 4,638,437*, Jan 1987.
- [8] Wagenmakers J H. Aircraft Performance in Take-Off and Landing. The Good News and the Bad News, *41st International Flight Safety Seminar, Flight Safety Foundation*, Sydney, December 1988.
- [9] Khatwa R. A Proposal for a Take-Off Performance Monitor (TOPM), *Technical Report*, University of Bristol, Department of Aerospace Engineering, October 1987.
- [10] Khatwa R. A Review of the Developments for a Take-Off Performance Monitor (TOPM), *Technical Report*, University of Bristol, Department of Aerospace Engineering, December 1987.
- [11] Khatwa R. Engine Monitors for Efficient Take-Off Performance Monitors, *2nd International Congress on Condition Monitoring (COMADEM 90)*, London, July 1990.
- [12] Khatwa R. Optimal Filtering of Sensor Signals for Take-Off Performance Monitors (TOPM), *17th Congress of the International Council of Aeronautical Sciences (ICAS)*, Stockholm, September 1990.
- [13] Khatwa R. *The Development of a Take-Off Performance Monitor (TOPM)*, PhD Thesis, University of Bristol, Department of Aerospace Engineering, January 1991.
- [14] Khatwa R. The Development of an Efficient Take-Off Performance (TOPM), *18th Congress of the International Council of Aeronautical Sciences (ICAS)*, Beijing, September 1992; also published as NLR TP 92226 L.
- [15] Verspay J J L, Erkelens L J J. Voorbereidingen voor Onderzoek naar een Take Off Performance Monitoring System TOPMS (Preparations for Research on a Take Off Performance Monitoring System), *NLR CR 90076 L*, 1990 (in Dutch).
- [16] Verspay J. Take-Off Performance Monitoring System Algorithm and Display Development, *AIAA Atmospheric Flight Mechanics Conference*, New Orleans, August 1991; also published as NLR TP 91395.
- [17] Verspay J., Khatwa R. Workstation Evaluation Of Three Take-Off Performance Monitor (TOPM) Display Types, *NLR CR 93202 L*, 1993.
- [18] SAE. Take-Off Performance Monitor (TOPM) System, Airplane, Minimum Performance Standard for, *SAE AS-8044*, Society of Automotive Engineers (SAE), Warrendale, PA, 1987.
- [19] Boeing. Takeoff Safety Training Aid, Boeing Commercial Airplane Company, Seattle, 1992.

The Development of SIMONA: A Simulator Facility for Advanced Research into Simulation Techniques, Motion System Control and Navigation Systems Technologies

S.K. Advani*

Faculty of Aerospace Engineering
Delft University of Technology
Delft, The Netherlands

ABSTRACT

The design and development of a new-technology six-degrees-of-freedom research simulator at the Delft University of Technology is described in this paper. This simulator will incorporate an advanced hydraulic motion system and a light-weight moving platform for outstanding dynamic performance. The primary design goal is to set a new standard for the fidelity of motion simulation. The simulator will become the core of the new International Centre for Research in Simulation, Motion and Navigation Technologies, or "SIMONA". Fundamental research in the SIMONA facility will be aimed towards the development of simulation modelling techniques, for the refinement of motion system control, and for investigations into pilot interactions in realistic navigation environments. This simulator, now under construction, can be configured to represent a wide variety of vehicles including fixed and rotary-wing aircraft as well as surface vehicles for land and sea operations. This results in a variety of multi-disciplinary research roles for the simulator.

INTRODUCTION

Society and the transportation industry are placing severe and new demands on the performance, safety, and environmental qualities of current and future transportation systems. The safety of transportation systems, the noise and effluent emissions generated (and resulting damages to ecosystems) are issues of growing concern to citizens, governments and industries. Since it is expected that manned vehicles will remain as the primary means of transportation, it is necessary to optimize the interface between the human controller with his limited capability, and the machine.

Vehicle-operator interactions, or Man-Machine Interfaces (MMI's), can be very well evaluated in moving-base simulators in which the environment in and around the vehicle can be accurately synthesized. For such research, the fidelity of the physical cues, especially the motion, are of utmost importance, as well as the accuracy of the mathematical models which describe the vehicle behaviour.

Simulation technology on the other hand must first be furthered in order that the most meaningful data can be derived from simulator trials. Therefore, new approaches to simulator hardware and software are proposed in the SIMONA program, with the "Basic Research Simulator" as the proof-of-concept hardware platform¹.

The Delft University of Technology (DUT) has played a unique role in the development of simulator technology over the past two decades, and has also been using simulators as instruments for human factors research. Development-related efforts have focused on motion system design², motion drive laws³ and hydraulically-driven sidestick controllers⁴. Hydrostatic bearings, which are used in every high-performance motion system to date are a result of a continuous quest to improve motion simulation fidelity. The research "platform" for these developments, a three degrees-of-freedom simulator, has also served to evaluate human motion perception studies⁵ and, more recently, the evaluation of "tunnel-in-the-sky" navigation displays⁶. Various airplane mathematical models have been developed and evaluated with help from this system. The application of parameter identification techniques have produced highly accurate representations in some cases⁷.

The simulator described above is located at the Faculty of Aerospace Engineering. Research in flight simulation and its related disciplines is of interest however to three Faculties at the DUT: Aerospace Engineering, Mechanical & Marine Engineering, and Electrical Engineering. Respectively, these groups have cultivated expertise in simulation systems and techniques, motion systems and control, and aircraft navigation/instrumentation systems modelling. Furthermore, strong cooperations in each of these disciplines exist with research partners from abroad.

SIMONA, the International Centre for Research in Simulation, Motion and Navigation, is an initiative which will merge new developments in these three areas in varied applications. The primary aim is in fundamental research into simulation techniques, and in developing human-machine interfaces for flight and non-flight vehicles. Conversely, the "National Simulation Facility", operated by the National Aerospace Laboratory NLR in Amsterdam, concentrates on applied or operational systems research. Respectively, these groups represent the fundamental, and applied sides of simulation

* Assistant Professor / SIMONA Project Leader,
Member AIAA

research in the Netherlands⁸.

SIMONA will address fundamental scientific research through international collaboration with twelve internationally-renowned research institutes (see Ref. 1). New developments will directly benefit those industries which produce simulator hardware, software, or make use of simulators for research and training. An international network between research and industry will result. Currently, the industries which formally participate in SIMONA are Hydraudyne Systems and Engineering, Fokker Space & Systems and Fokker Control Systems. Other industrial partnerships, both Dutch and foreign, will be disclosed when negotiations have been finalized.

SIMONA will also work closely with the Delft Institute for Advanced Studies in Aerospace Engineering, or "DIASA", a new initiative for post-graduate research. Cooperation in this respect will focus on courses in simulation techniques, parameter identification and man-machine systems.

RESEARCH GOALS:

The two main purposes of SIMONA are (1) to improve existing simulation technology and develop new techniques which provide the maximum possible simulation realism at the operator (or pilot) position, and (2) to apply use a multi-vehicle simulator to investigate fundamental man-machine interaction issues. The Context Diagram of the SIMONA environment is shown in Figure 1. The three disciplines which comprise SIMONA - simulation, motion and navigation - are described below.

1. MATHEMATICAL MODELS FOR SIMULATION

Fixed-wing aircraft

Modern airplane simulation mathematical models, especially those used in airline training simulators, often roughly approximate or even neglect certain physical phenomena due to the complexity of determining realistic figures for their dynamic properties, and also due to the high computational load generated by these effects. Conversely, the inclusion of these, can significantly increase cue fidelity. It is felt that the following areas require special attention:

- ❑ aircraft structural flexibilities
- ❑ responses due to gusts, thus nonstationary aerodynamic loads interacting with structural deformations
- ❑ separated flows (buffet, effects due to spoilers, speed brakes, etc.)
- ❑ motor dynamics
- ❑ ground effects
- ❑ ground handling
- ❑ aircraft Active Flight Control (AFC) system design

One of the current emphases at DUT is the accurate representation of vehicle structural flexibilities in response to external forces, or aeroelasticity. The magnitude of such

phenomena is related to the flexibility of the vehicle. To meet long-range market demands with requisite efficiency, airplane manufacturers are developing vehicles with high aspect ratio, highly-flexible wings and generally incorporating light-weight constructions throughout (e.g. Boeing 777, Airbus A340, etc.). This also applies to the fuselage and other primary structures. For this category of aircraft, the frequencies of the first-order wing bending mode, "flapping", come close to those of the rigid body modes, such as the short-period motion. Therefore, the wing and fuselage flexibilities contribute substantially to the vehicle dynamics⁹, underscoring the need for accurate yet computationally efficient real-time flexibility models.

The development of these techniques requires a multi-disciplinary effort, involving aerodynamics, flight mechanics and structural mechanics. A proposed system configuration is shown in Figure 2. Here, the aircraft Flight Control System (FCS) drives an unsteady time-domain panel-methods code which then interacts with the structural flexibility and rigid-body dynamic models. The latter will be replaced by Navier-Stokes solutions once these are developed.

Since a primary aim of flight simulation is to recreate the same flying qualities as the aircraft, one application of the flexible models is to represent the real handling qualities in the simulator and to generate the motion cues caused also by the flexibilities. However, it is also vital to take the time-varying structural deformations into account during the design of AFC systems for such aircraft, since it is necessary for an AFC system to have pre-determined knowledge about all of the aircraft's natural characteristics. Therefore, the structural flexibilities must be accounted for to manage flutter mode control, ride control, and flexible mode control in the Active Flight Control system.

A real-time aeroelastic simulation program is now under development. This will be based on the results of an unsteady Computational Fluid Dynamics (CFD) code, interacting with a Computational Structural Mechanics (CSM) program. Such a mathematical model would theoretically represent an infinite number of flexible modes, plus the six degrees-of-freedom of the rigid-body dynamics, resulting in a system with infinite degrees-of-freedom. Because it is uninteresting and also very complex to accurately simulate the very high-frequency components of the structural flexibilities, a generic "reduced-order" model replicating only the lower frequency range is now being developed. This essentially cuts off the calculation of unnecessarily high frequencies. For real-time motion-based simulation, tradeoffs which ensue the order reduction would in any case be unnoticed due to two factors: the physical capabilities of the motion system, and high-frequency human motion perception thresholds.

The lowest structural flexibility frequency, representing the static deformation under steady state, is also needed to define very accurately the performance of the flight vehicle under given conditions of weight, load distribution, etc. The

responses of flexible aircraft structures to turbulence, interactions with ground effect, loads will also be simulated with these models. In addition, ground contact dynamics which are influenced by flexibilities will be incorporated.

Spaceplane dynamics

The DUT will investigate the dynamics and handling qualities of spaceplanes throughout their flight regimes. Unsteady aerodynamic (CFD) codes will be used to set up the models and to estimate their parameters. Structural flexibility and engine dynamics modelling is also being developed. In hypersonic flight, it is possible that structural deformations due to thrust, aerodynamic loads and heating can distort the shock wave patterns around the vehicle and, in turn, the performance may be affected. Furthermore, interactions can occur between the low-frequency rigid-body motions (such as speed and pitch), and the higher-frequency structural flexibilities.

Eventually, these spaceplane mathematical models will be used in pilot-in-the-loop simulations to investigate the FCS requirements and the design of the flight instrumentation for these vehicles.

Rotorcraft Dynamics

Foundation-level research has begun to develop flexible-structure blade-element models for rotors which include nonlinearities (stall, reverse flow, high Mach number). Aerodynamic models for blade/body/tail rotor interactions are also being developed using unsteady aerodynamic principles.

The responses to turbulence and the interaction with ground effect will also be modelled for future investigation in the simulator. Advanced rotorcraft FCS's will also be investigated.

Human motion perception research

The identification of human perception thresholds and vestibular system dynamics is necessary for the design of simulator hardware and software. Emphasis will be placed on modelling and identification of the following:

- ☐ Visual-vestibular interactions in motion perception
- ☐ Vestibular system dynamic properties
- ☐ Thresholds for mismatch of visual and vestibular motion cues and motion sickness¹⁰
- ☐ Disorientation

Human visual perception research

Fundamentals into human visual perception can be well-identified in a research simulator. Principles such as object identification and reading as functions of colour, contrast and shape can be investigated by using reprogrammable display software. A special display software system has been developed for the evaluations of these and other man-machine issues¹¹.

The SIMONA facility will also examine human pilot behaviour during events which require the timing of actions, such as the

initiation of the flare maneuver during the landing phase¹². The understanding and modelling of these perception processes is of importance to the design of flight and non-flight vehicles and simulators.

2. MOTION SYSTEMS AND HYDRAULICS RESEARCH

The underlying aim of the motion control research is to control the dynamics of the pilot position(s) such as to experience dynamic behaviour which strongly resembles reality. This should hold for the entire range of motions experienced in a vehicle; that is to realistically present low-frequency long-stroke maneuvering motions, and also the sharp, high-frequency vibrations due to turbulence, structural deformations, or ground-contact oscillations.

One of the primary aims of SIMONA in the area of motion systems research is to achieve a *virtual zero* motion time delay over the normal operational bandwidth of the six-degrees-of-freedom motion system. This is accomplished in two ways - the hardware provided is highly advanced (see the "Hardware" discussion below), and secondly new robust motion system control techniques are introduced. Robustness is key to the controlling of the continuously varying nonlinear dynamics of the motion platform.

In the area of motion control, there are three "levels" which are under development, to eventually adjust the entire motion control algorithm to the specific properties of the system, acting as a six-degrees-of-freedom motion system. The system schematic, Figure 3, illustrates this concept. In effect, we recreate the motions by forcing the natural eigenfrequency behaviour of the platform at the location of the pilot, a principle called "eigenvalue assignment". The six-degrees-of-freedom motion system allows the necessary bandwidth of control. A control system architecture which exercises the dynamic control over the modeshapes of the platform so that the pilot experiences the vehicle structural dynamics as realistically as possible is being developed. The proposed three levels of control necessary to achieve this are as follows:

a) Modelling and control of individual hydraulic actuators

Advanced control law design begins with the determination of the system properties. In the case of a motion system, the most fundamental element is the cylindrical hydraulic actuator. First, the properties of this are modelled, including the hydraulic-line dynamics, valve properties, leakage flows, and so on. The resulting nonlinear simulation model is now being experimentally validated for a prototype actuator which closely resembles the hardware which will be used in our simulator. Next, control strategies are being developed with which the bandwidth, output error, robustness, and linearity are verified. These control strategies so far include linear techniques such as classic methods, robust control (H_∞ , μ), as well as non-linear compensation methods. These may be extended to multivariable controller design later on.

It is expected that the dynamic properties of hydraulic actuators can be significantly improved by using such strategies while retaining very favourable robustness properties. This is accomplished by taking into consideration the parameter uncertainties and system variations.

Some practical examples of these phenomena, in a simulator motion system, are:

- ☐ variations in static loads (number of on-board occupants, new hardware configurations)
- ☐ dynamic variations in actuator loads throughout the operational envelope
- ☐ internal system variations: leakage flows associated with hydrostatic bearings, oil temperature fluctuations, degraded (contaminated) servo valve performance, contaminated oil.

With increased performance, robustness in a motion system can also reduce its maintenance time. The strategy illustrated can also indicate clearly where hardware and hence maintenance problems exist.

b) Dynamics of the coupled six-degrees-of-freedom motion system

In the previous level of control, the individual actuators were controlled as independent units. The next step is to further utilize the model knowledge built in the previous phase and to coordinate the six actuators such that the desired six degrees-of-freedom motion becomes possible under a high-performance closed-loop feedback network. The hydraulic actuators in fact become "white boxes", with their state of operation being fully known by this second-level controller. Thus, a general controller is developed to account for the reflected masses encountered on each motor, and which compensates for all instantaneous interactive loads.

A first approach will investigate linear and nonlinear multivariable strategies for motion platform control to compensate for platform loads and dynamics.

In a second approach, while the simulator is in use, a parallel model of its dynamics is run in real-time. This model is driven by the same signals which drive the motion system actuators, and measures the actual load on the motion system. Predictive control implemented in the parallel model can then be applied to the motion platform, providing a deterministic outcome.

c) Motion-drive algorithms

To complete the motion control scenario, the transformation of aircraft motions to the limited simulator displacements is necessary. This is called the motion-drive, or washout algorithm. Due to the large compromises which must be made in trying to represent the physical motion cues within the mechanical limitations of the simulator motion system, there usually exist false cues, or motion errors. Washout also tends to reduce the sensation of false cues by tricking the human vestibular system and masking certain motions which

are necessary to limit the displacement of the motion platform. Recent investigations into adaptive motion-drive algorithms have shown that real-time alterations of motion control signal gains, in order to minimize a cost function, can reduce the onset of the false cues¹³.

A second strategy however will focus on using aircraft-dependent parameters to objectively define the control parameters. Furthermore, data derived from the human motion perception research will also support the development of refined motion-drive algorithms. In particular, the thresholds for the sensation of linear motion are as of yet vaguely defined and will be concentrated upon. Results from recent and ongoing human motion perception research¹⁰ in SIMONA will be incorporated into future motion-drive algorithms.

Sidestick Controllers

Most aircraft which fly today with sidesticks as primary controllers provide second-order passive stick properties which are usually not changed during the flight. By introducing hydraulic coupling into a sidestick, it is possible to alter the properties in software, in a way similar to simulator control loading systems. Properties can be varied according to the task (i.e. stiffness proportional to airspeed). A servo-driven sidestick can improve the pilot's situational awareness: by driving the roll position by roll rate, a pilot can provide a corrective control feedback to compensate for lateral disturbances, resulting in "active control", an area of ongoing research at DUT. Neuro-muscular modelling of the human arm is also used to identify the gains required in such configurations¹⁴.

3. NAVIGATION AND AVIONICS SYSTEMS RESEARCH

The navigation research area of SIMONA is sub-divided into two parts; the modelling and identification of radio navigation system properties for real-time simulation, and the development and evaluation of the fundamental properties of advanced cockpit displays.

Radio navigation systems modelling and simulation

It is of concern to the aircraft avionics designer, the pilot, and regulating authorities to know how the pilot (or auto pilot) will react to errors in radio navigation systems, particularly during the landing phase. Prior to in-flight evaluations, the navigation equipment must be modelled, either as simple receiver devices such as tracking loops, or as highly-integrated systems based on e.g. GPS, MLS, ILS and AHARS. Research is directed at the following:

- ☐ Optimum tracking loop design in radio nav aids for maneuvering aircraft
- ☐ Optimization of Kalman filters in integrated/hybridized nav aids for all flight phases
- ☐ Pilot reactions to infrequent, yet large and realistic navigation errors
- ☐ Multi-path and shadowing effects on position with GPS

- and MLS position determination
- ❑ Attitude determination of aircraft with GPS in the presence of structural flexibilities
- ❑ The verification of approach and missed-approach performances

A close cooperation with the Avionics Engineering Center of Ohio University has led to the development of signal models which include multipath, D/GPS (Differential GPS), data link and D/GPS reference receivers¹⁵. Radio navigation systems research is not restricted to modelling and simulation alone; investigations into the feasibility of various integrated/hybridized radio navigation systems have also been a recent undertaking. This group also focuses on the rudiments of radio navigation receivers for GPS with Receiver Autonomous Integrity and Signal Monitoring (RAISIM) capabilities¹⁶.

Evaluation of modern flight instrumentation displays

Aerospace technology is very rapidly changing in areas related to avionics, navigation systems, digital flight control, and electronic aircraft systems management. Before modern avionics and navigation systems can be implemented into service however, their fundamental properties and hence the Man-Machine Interface must be thoroughly evaluated. Once again, the research simulator can be used to synthesize the cues which a pilot would experience when interpreting or interacting with a display system. One recent and notable activity deals with the evaluation of future four-dimensional navigation system displays for aircraft.

Due to the increase in air traffic, the demand for tighter control of commercial aircraft is increasing. Current Air Traffic Control (ATC) procedures provide regulation over very short time spans, due to the lack of accurate information regarding the future trajectories of flight vehicles. This results in a new set of constraints every few minutes, which increase with the volume of air traffic.

Since trajectory information is already present in the aircraft Flight Management System (FMS), such knowledge by ATC would provide control over greater time spans. One means of managing aircraft through airspace is by integrating ATC and FMS information, and presenting a combined navigation/primary flight display in the cockpit. As opposed to current tactical control, this proposed system would provide long-term strategic planning capabilities for both ATC and the flight crew. Presented in a perspective form, a candidate "tunnel-in-the-sky" display is shown in Figure 4. These displays are generated by the D³S system, a hardware-independent software package for the creation of instrumentation displays¹¹.

In order to improve the flying characteristics while operating these displays, it is necessary to provide predictive information of the aircraft state to the pilot. This is given in the form of a small aircraft symbol, driven by an algorithm which determines the attitude and location some seconds ahead. Predictors have been developed for tunnel-in-the-sky

displays¹⁷ and preliminary evaluations of these have been carried out in the current simulator facility⁸.

The tunnels-in-the-sky, which will become available in future aircraft displays, are mathematically-determined paths which the pilot is advised to follow. However, these are built from information from reference radio navigation signals and the aircraft flight management system. The accuracy of these depends on that of the signal reaching the aircraft. Since the signal source will likely be D/GPS, the position and altitude signal updates nominally at one cycle per second. Displays however must be refreshed at a rate at least thirty times greater. A technique based on Kalman filtering, now under development, aims at minimizing the error associated with the low-update D/GPS signals.

SIMONA HARDWARE

The core of the SIMONA hardware is an advanced simulator system, called the "Basic Research Simulator", (BARESIM). This is a multi-purpose device capable of representing the cockpit or driver station of many vehicle types.

It should be clear from all the above that this simulator will have to fulfil a multitude of needs and will require quick turn-around times. The BARESIM features modular hardware (control manipulators, instrumentation) and software (outside-world display, instrumentation, control loading). The design principles are further explained in Reference 18.

Six-degrees-of-freedom motion system

The kinematic motion envelope of BARESIM is given in Table 1. (Dynamic properties are currently being established). Six double-concentric actuators (Figure 5) are joined together to create the synergistic motion system. These will have a stroke of 1.15 m with an additional 0.05 m safety buffer on each end. Note that for either extension or compression the piston areas are equal, allowing the use of a small symmetric servo valve having linear flow in both directions: When the flow direction is reversed, there is no discontinuity at the zero-flow point. The result is smoother operation and a relaxed control requirement.

The double-concentric design also inherently provides the actuator with high lateral stiffness. The actuators proposed for BARESIM have been preliminarily analyzed as having a 17 Hz lateral eigenfrequency, in this case the first-order natural bending frequency. A high lateral stiffness is necessary for the motion system's structural stability, and improves its response to high-frequency control inputs.

Motion platform

The simulation of a number of vehicles is a major design goal of this simulator. Furthermore, human motion and visual perception modelling and identification will be of particular emphasis in SIMONA research. Due to the sensitivity of the human senses, this requires the highest possible accuracy in the dynamic properties of the motion system so that the

results are not polluted by parasitic noise. In particular, linear motion (combined heave/surge) will be studied, a motion which without careful hardware and software design is prone to signal errors toward the ends of the strokes.

In order to achieve the high-fidelity dynamic response from the advanced motion system, the moving platform of the BARESIM is designed for a low total mass, low centre-of-gravity, and a minimum total inertia. The philosophy behind this is that actuator control can be improved by lowering the reflected loads on the motion system. The amount of compensatory control is thereby minimized and parasitic noises reduced. With prediction and the use of advanced mathematical control engineering, it becomes possible to realize a response with virtual zero time delay over a very wide frequency range (0-15 Hz). This capability is not offered in simulators today due to both hardware and control software limitations. BARESIM will be a proof-of-concept platform to demonstrate that light-weight platforms offer significant advantages to simulation.

Dynamic simulation of motion platform

To demonstrate the importance of the simulator payload mass and the location of its centre of gravity, a parametric study has been conducted using a program to simulate the behaviour of a simulator motion system itself. This detailed software package accounts for the system geometry and all internal properties of the actuators¹⁹.

In the simulation trials presented below, the servo valve input data was either generated from information from the supplier, or was estimated. The payload is approximated as a sphere with a diameter of 3.0 m and the mass evenly distributed within. For a comparative parametric study, exact properties were not necessary. These will however be incorporated in future evaluations. A block-function input signal (zero, step-up, step-down, zero) commands the desired acceleration of the platform.

The actuator control feedback diagram, shown in Figure 6, illustrates that only acceleration and position feedback signals, acting as internal damping terms, are used; it was found that neglecting either of these would severely destabilize the actuator. Apart from this feedback, the actuators are in fact open-loop systems responding to a desired acceleration input signal. No other form of compensatory signal to correct for overshoots or phase lags is applied, and the internal damping terms are not tuned for the best possible damping. In an operational motion system, tuning must take place, and the remaining undesired motion must be compensated by signals external to the actuators. The following discussion therefore illustrates how the relative degree of such compensation is influenced by two design factors: payload mass and vertical location of the mass centre.

In the following, only the results for the x-direction (surge) acceleration input signal are discussed.

Influence of payload mass variations

In the first evaluation, the centre of gravity is placed at the centroid of the upper bearings and has no lateral or longitudinal offset. The responses to the same input signal are then compared for a payload mass of 6000 kg, 4000 kg, and again for 2000 kg. Note that for most commercial training simulators the mass exceeds 6000 kg.

The desired (input) and achieved translational x-accelerations are shown in Figure 7a. Clearly, the mass plays a profound role. In particular, the response time (the delay) increases with mass. Accordingly, the amount of compensation that is necessary, in order to eliminate these undesirable effects, would also increase with mass.

Figure 7b shows that the pitch acceleration, the prime parasitic motion when a pure x-acceleration is desired, is strongly influenced by the payload mass. The largest excursions occur when the platform has reached its forward limit and a reverse acceleration is commanded. At this point (t=2 seconds) the actuators experience very high dynamic (reflected) loads, leading to the large overshoots in q-dot and also in q-dot.

Influence of payload centre-of-gravity location variations

In a second investigation, the payload mass is maintained at 6000 kg while the vertical location of the centre of gravity is incremented in three steps: 2.0 m, 1.0 m, and 0.0 m. The latter represents a centre of gravity coincident with the upper gimbals' geometric centre. Training simulator centres-of-gravity are located approximately two metres or more above the centroid, depending on the hardware that is installed.

Figure 8a shows that the response time (to achieve the desired 1 m/s² acceleration in surge) does not change significantly with the centre-of-gravity location. The amplitude error or "overshoot" however increases considerably. The position in time of the maximum overshoot also increases with higher centre of gravity, due to the higher actual loads on the actuator as the platform nears the forward extremity of the motion envelope.

Note that even when the centre-of-gravity is coincident with the upper gimbal plane, there still exists a parasitic response; that is, pitch coupling due to surge (Figure 8b). This is due to the rotational moment caused by the application of the initial actuator force required to cause translation in the x-direction, and also the inertias of the actuators themselves. This further promotes the suggestion that minimizing the payload mass is necessary. There will always exist parasitic motions, however their effects on the system behaviour and on the compensation required can only be reduced through careful design of the motion platform.

Optimizing the mass distribution of the simulator platform

The platform of BARESIM is actually a structural shell with the requirements of the cockpit stations integrated within. Instead of being mounted on top of the motion system as most commercial training simulators are, the gimbals are

attached to the outside, about one-third the way from the lower surface. Although this restricts the space available in the cockpit (unless a very large motion system is used), careful use of the interior space yields a design that is generally superior.

The BARESIM motion platform is shown in Figure 9. This particular candidate, the latest version, is to be constructed from continuous carbon-fibre reinforced polymers, materials which are extensively applied at the Faculty of Aerospace Engineering. It is estimated that the structural mass of the motion platform, excluding on-board systems, will be 175 kg. As a result of the use of these advanced materials and also due to fundamental design differences, this represents substantial weight savings over conventional steel motion platform constructions. The materials suggested offer high stiffness combined with a cost-effective manufacturing solution. Other candidate platform materials and geometrical shapes are also being evaluated to compare their relative merits.

Wide-angle visual display system

A wide-angle projection-based visual display system will generate accurate visual scenarios for all simulation research environments. Again, this system will be integrated with the platform/cockpit structure to optimize the weight distribution. Selected optical and projection hardware components will be purchased, while the primary structural elements will be designed using carbon-fibre-reinforced polymers. Although any on-board display system raises the weight and centre-of-gravity location, the use of modern materials in an integrated structure reduces this contribution as much as possible.

The display generation system will be based on three high-resolution CRT projectors. High-Definition Television (HDTV) and Liquid Crystal Display (LCD) projectors are being considered as candidates for new technology development. A wide-angle projection-based collimation system was chosen because it offers the possibility to work with a large uninterrupted field-of-view, and is less sensitive to pilot head position. In BARESIM, the research stations will be arranged as side-by-side (transport aircraft), or will be centred in the x-z plane. With the display system selected, no modifications to the hardware will be necessary when changing the research environment. The proposed mirror cell/platform integration is shown in Figure 10.

Cockpit interior

The cockpit is designed for a maximum of three occupants - two pilots, one experimenter/observer. Various generic vehicle research stations are designed to simulate the physical cues and control aspects:

Transport aircraft station

The transport cockpit shown in Figure 11, an artist's impression, shows the spaceplane research station configuration. Five programmable displays will generate the EFIS, Tunnel-in-the-Sky, Navigation Display, EICAS, etc. Removable control manipulators, both yoke and sidesticks,

will be hydraulically driven. These will be coupled together through software only, reducing the mass and mechanical complexity. Throttle controls will represent four jet engines, or two turbo-props, each with rpm and throttle control. Rudder pedals with brakes will also be hydraulically loaded. A current research project will determine which controls can be replaced by electrical control loading devices for minimal total weight penalties.

Helicopter station

The transport aircraft cockpit hardware can be modified to represent a helicopter environment by installing the requisite control manipulators (cyclic and collective pitch). With ease of display programmability, helicopter instrumentation can be created, and new display presentation techniques developed. Due to the need for increased vertical field-of-view in helicopter operations, monitor-based collimating "chin windows" will be attached to the platform structure (in a later phase).

Road-vehicle station

Modules representing automobiles, trucks, trains, etc. can also be integrated in BARESIM, since all cockpit sub-assemblies are designed to be quickly removed.

Motion perception research station

The translational motion envelope of the six-degrees-of-freedom motion system can be extended by installing in the cockpit chamber a sliding seat which is driven by a linear actuator. Such a configuration is shown in Figure 12, with a 0.6 metre hydraulic actuator. In conjunction with the normal motion envelope, this would permit larger displacements in the x-z plane for the purposes of low-acceleration linear motion studies on human subjects.

JOINT RESEARCH WITH THE "NATIONAL FLY-BY-WIRE TESTBED" NFT AIRCRAFT

The Delft University of Technology and the National Aerospace Laboratory NLR jointly own a new Cessna Citation II aircraft which will be configured as a fly-by-wire test vehicle. SIMONA will therefore have access to a high-performance aircraft with a large flight envelope for the identification of mathematical models for simulation.

Conversely, research conducted in the flight vehicle can be pre-evaluated in the flight simulator, for example the evaluation of 4D navigation displays.

Active Flight Controls (AFC) will be a major area of study with this new vehicle, with emphasis on the following topics:

- ☐ Baseline (classic) control laws evaluation
- ☐ Development and evaluation of new control theories
- ☐ Active/passive manipulators
- ☐ Degraded flight control system
- ☐ Smart actuators

CONCLUSIONS

SIMONA, The International Centre for Research in Simulation, Motion and Navigation Technologies at the Delft University of Technology will offer researchers with a very advanced facility for fundamental research. Embedded in this institute is the multi-purpose and high-performance Basic Research Simulator. It is expected that this facility shall become operational in 1995.

The technologies incorporated in this facility, and those yet to be developed will have significant scientific and economic benefits to simulation, transportation and control industries. The research conducted in SIMONA will also support new methods of improving transportation system efficiency, safety, and environmental qualities. The Basic Research Simulator is shown in artist's concept in Figure 13.

ACKNOWLEDGEMENTS

The author would like to thank his colleagues at the Delft University of Technology for their contributions while preparing this report. Appreciation is extended to the Netherlands Organization for Scientific Research NWO for partly supporting the projects mentioned herein. The many participants of SIMONA are also thanked for stimulating and contributing to scientific research.

REFERENCES

1. Advani, S.K., "The Basic Research Simulator and the Industrial and Aerospace Community: Opportunities for Cooperative Research". TU Delft Report LR-662, October 1991.
2. Viersma, T.J., Baarspul, M., "Development and Application of a Moving-base Visual Flight Simulator, Including the Design of Hydraulic Actuators with Hydrostatic Bearings". ICAS-80-9.2, 12th Congress of the International Council of the Aeronautical Sciences, Munich, 1980.
3. Baarspul, M., "A Review of Flight Simulation Techniques". From Progress in Aerospace Sciences, Vol 27, No. 1, 1990, ISSN 0376-0421/90.
4. Hosman, R.J.A.W. and Vaart, J.C. van der, "Active and Passive Side Stick Controllers: Tracking Task Performance and Pilot Behaviour". AGARD CP-425.
5. Vaart, J.C. van der, "Modelling of Perception and Action in Compensatory Manual Control Tasks". Ph.D. thesis, Delft University of Technology, December 1992.
6. Theunissen, E., "A Primary Flight Display for Four-Dimensional Guidance and Navigation - Influence of Tunnel Size and Level of Additional Information on Pilot Performance and Control Behaviour". AIAA Flight Simulation Technologies Conference, Monterey, August 1993.
7. Baarspul, M. and Mulder, J.A., "The Composition of Flight Simulation Models: Flight Testing Versus DATCOM Techniques". AIAA-90-3120-CP. (From AIAA Flight Simulation Technologies conference proceedings, 1990).
8. Vermeulen, W and Advani, S.K., "Fundamental and Applied Simulation Research in the Netherlands to the Benefits of Aviation Safety". 1st International Congress on Safety of Transportation, Delft, October 1992.
9. Waszak, M.R., Davidson, J.B., Schmidt, D.K., "A Simulation Study of the Flight Dynamics of Elastic Aircraft". NASA Contractor Report 4102, December, 1987.
10. Hosman, R.J.A.W. and Steen, H. van der, "False Cue Detection Thresholds in Flight Simulation". From AIAA Flight Simulation Technologies conference, Monterey, August, 1993.
11. Theunissen, E., "D³S: The DELPHINS Display Design System". From AIAA Flight Simulation Technologies conference, Monterey, August, 1993.
12. Advani, S.K., Vaart, J.C. van der, "What Optical Cues do Pilots Use to Initiate the Landing Flare? Results of a Piloted Simulator Experiment". From AIAA Flight Simulation Technologies Conference, Monterey, August, 1993.
13. Jansen, E., "Implementation and Evaluation of Nonlinear Filters in the APCOFIL Algorithm". Delft University of Technology masters thesis report, September 1992.
14. Paassen, R. van, "Human Controller Response to Visual and Proprioceptive Inputs in a Control Task with a Side Stick". 10th European Annual Conference on Human Decision Making and Manual Control, Liege, Belgium, November 1991.
15. Braasch, M., "A Robust D/GPS Signal Model to Support Real-Time and Fast-Time Aircraft Simulation". In ION CPS-93, Salt Lake City, Utah, Sept. 1993.
16. Nee, D.J.R. van, "Optimum D/GPS receiver structures", DSNS '93 - Second Int'l Symposium on Differential Satellite Navigation Systems, Amsterdam, 1993.
17. Mulder, M., "Aviation displays and flight-path predictors". Delft University of Technology master's thesis report, November 1992.
18. Advani, S.K. and Baarspul, M., "Design Principles of the Basic Research Simulator". 18th Congress of the International Council of the Aeronautical Sciences, Beijing, September 1992.
19. Kuit, J.J., "A Nonlinear Simulation Model of a Six-Degrees-of-Freedom Motion System". Delft University Thesis Report, July 1991.

	Pitch	Roll	Yaw
Angular limits (deg)	+ 25, - 24	± 27	± 43

Table 1. Total simulator motion displacements

	Vertical (heave)	Lateral (sway)	Longitudinal (surge)
Total translational limits (m)	1.34	2.04	2.24 (2.84*)

* denotes extended motion envelope when sliding seat is installed

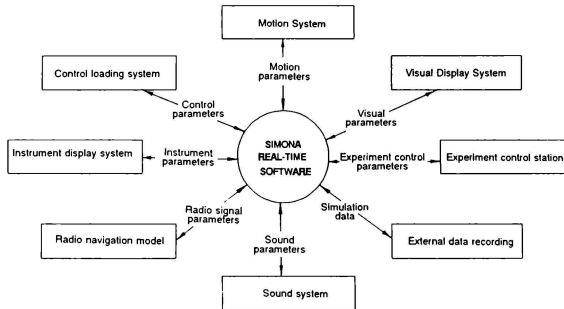


Figure 1. Context Diagram of SIMONA simulation software

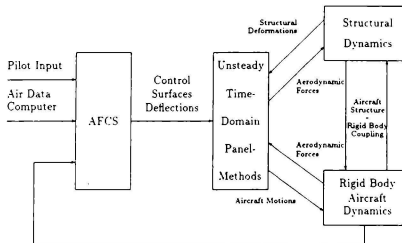


Figure 2. Proposed interactive aerodynamics-structural dynamics software

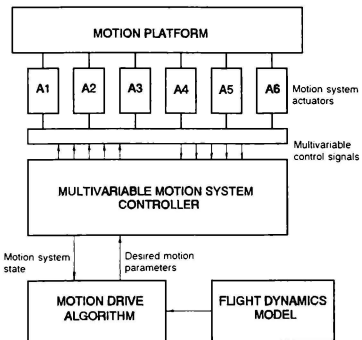


Figure 3. Motion control system schematic

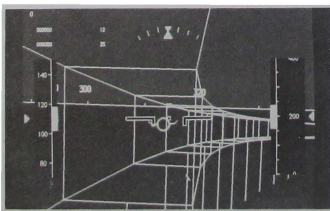


Figure 4. Perspective primary flight display for four-dimensional guidance and control

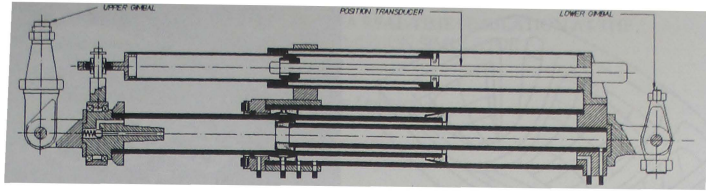


Figure 5. Double-concentric hydraulic actuator

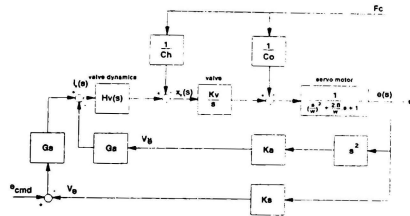
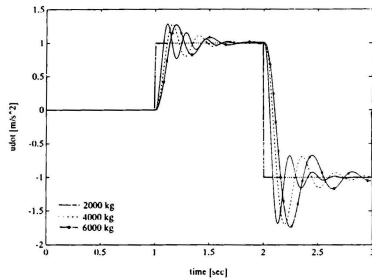
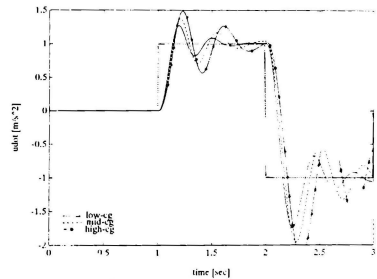


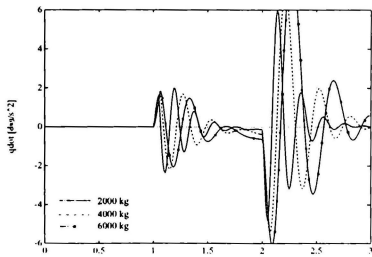
Figure 6. Internal control feedback for hydraulic actuator



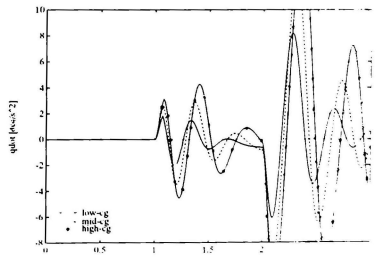
7a. Surge acceleration response



8a. Surge acceleration response



7b. Pitch acceleration response



8b. Pitch acceleration response

Figure 7. Motion platform dynamics in surge - variations in platform mass (low c.g.)

Figure 8. Motion platform dynamics in surge - vertical variations in platform c.g. location (6000 kg)

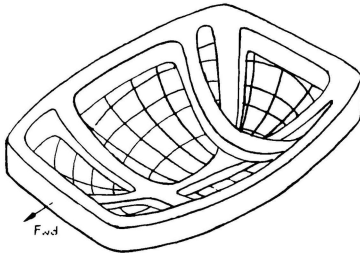


Figure 9. Composite materials motion platform lower structure

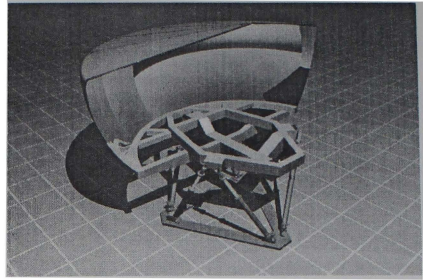


Figure 10. Platform/mirror cell integration

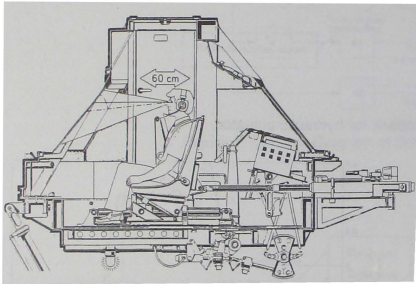


Figure 12. Sliding-seat arrangement for human motion perception research

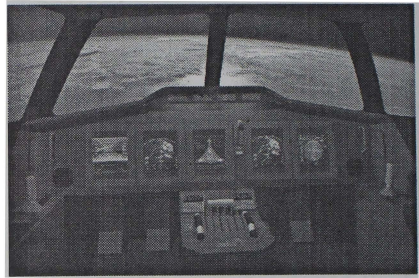


Figure 11. Transport aircraft cockpit (spaceplane configuration, artist's impression)

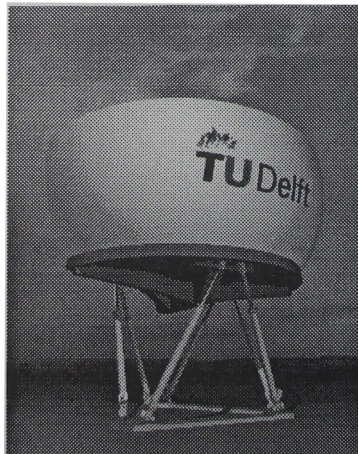


Figure 13. Basic Research Simulator (artist's impression)

DEVELOPMENT AND OPERATION OF A REAL-TIME SIMULATION AT THE NASA AMES VERTICAL MOTION SIMULATOR

Christopher Sweeney
Senior System Engineer

*Shirin Sheppard **
Senior System Engineer

Monique Chetelat
Senior System Engineer

SYRE, A Division of SYSCON Services, Inc.
NASA Ames Research Center
Moffett Field, California

Abstract

The Vertical Motion Simulator (VMS) facility at the NASA Ames Research Center combines the largest vertical motion capability in the world with an flexible real-time operating system allowing research to be conducted quickly and effectively. Due to the diverse nature of the aircraft simulated and the large number of simulations conducted annually, the challenge for the simulation engineer is to develop an accurate real-time simulation in a timely, efficient manner.

The SimLab facility and the software tools necessary for an operating simulation will be discussed. Subsequent sections will describe the development process through operation of the simulation; this includes acceptance of the model, validation, integration and production phases.

Introduction

The Ames Vertical Motion Simulator, shown in Figure 1, is a high-fidelity, piloted, six degree-of-freedom, real-time flight simulator. A variety of aircraft are studied including rotorcraft, STOVL aircraft, tilt rotor technologies and the Space Shuttle Orbiter. Research is conducted in the areas of flight controls, propulsion controls, guidance systems, cockpit displays and handling qualities. Other areas of interest include nap-of-the-earth flight, human factors issues, system failure evaluations and approach and landing studies.

The VMS facility attracts researchers by providing the ability to conduct high fidelity research in complex systems. Another valuable asset is the ability to modify software during the production session with minimum turn-around time. The hardware and software components of the facility allow

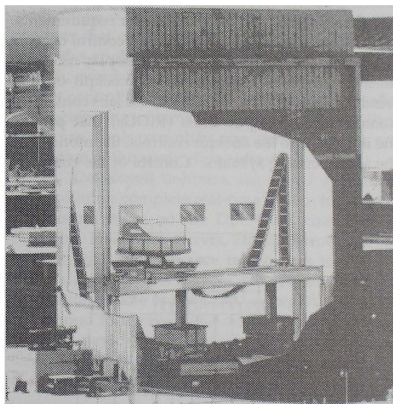


Figure 1 A Cutaway View Of The VMS Tower

engineers to run a successful real-time man-in-the-loop simulation every five to six weeks. This paper will focus on the methodology employed to achieve this compact simulation schedule at the VMS.

Initially, this paper will discuss the facility in terms of the hardware and software necessary to build and execute a real-time simulation. Subsequent sections will cover the application of these components in the chronological development, integration and production phases of a simulation.

SimLab Facility

A simulation at the Ames Research Center Flight and Guidance Simulation Laboratory (SimLab) involves connecting a series of diverse components

Each lab is equipped with a control console, real-time data recorders and video display monitors, normally used as repeaters of the cockpit out-the-window view. Electronic racks in the labs contain the Remote Input/Output Units (RIOUs) that provide the interfaces to the cockpit controls, the motion and the sound cueing systems. Control of the computer

SimLab has three host computers for running simulations. A VAX 9000 Model 210, capable of 40 VUPs (VAX Unit of Performance), and a VAX 6000 Model 610, capable of 32 VUPs, are the primary hosts. A VAX 4000 Model 6000, a 30 VUP machine, is used for less computationally intensive simulations or for simulation development. Several MicroVAX-III computers and a MicroVAX 3600 are also available for development. In some simulations, these MicroVAX computers are used as attached processors that run programs that cannot be run on the host. Such a case is Dynapath (pathway-in-the-sky) which is an asynchronous, non-deterministic program.



Currently, there are two computer image generators available to display the out-the-window scene. One is the Singer Link DIG-1 and the other is the Evans and Sutherland CT5A. The DIG-1, a vertical raster scan system, has four channels and two eye points. One eyepoint can depict an out-the-window view and the other a chase plane view. The CT5A system has a programmable scan capability with three channels and one eyepoint which is used for an out-the-window view.

Silicon Graphics 4D/310 VGXT IRIS workstations provide the HUD and the HDD graphics. The workstations generate moving map displays, cockpit instrument displays and aircraft status displays. Some simulations utilize the IRIS to distribute the workload of computationally intensive programs. In a recent simulation, a database residing on the IRIS containing terrain height information and obstacle positions was used to simulate a sensor for height above terrain, obstacle detection and guidance control.

The six degree-of-freedom motion simulator has a vertical displacement of 60 feet and a horizontal displacement of 40 feet in the long axis and 8 feet in the short axis. The system is comprised of three rotational axes, driven hydraulically, and three translational axes; one of which is driven hydraulically. The two larger translational axes are driven electrically. A unique feature of the VMS is that the two horizontal translational axes can be interchanged by attaching the yaw actuator in either of two locations, ninety degrees apart. Thus, the cab can be given the large displacement in the lateral or longitudinal direction depending upon the research requirements. Table 1 describes some of the performance limits of the motion system.

Table 1 VMS Motion System Performance Limits

Axis	Displacement (±)	Velocity (±)	Acceleration (±)
Longitudinal	4 ft	5 ft/s	16 ft/s/s
Lateral	20 ft	8 ft/s	13 ft/s/s
Vertical	30 ft	16 ft/s	22 ft/s/s
Roll	0.31 rad	0.9 rad/s	4 rad/s/s
Pitch	0.31 rad	0.9 rad/s	4 rad/s/s
Yaw	0.42 rad	0.9 rad/s	4 rad/s/s

Software Tools

The MicroTAU (μ TAU) real-time operating system forms the backbone for all simulations run at SimLab. Part of μ TAU, the Computer Aid for Simulation Programmers, Researchers and Engineers (CASPRE) real-time debugger is used in all simulations. The BASIC library contains various routines that can be used in the model depending upon the application. Configuration of the μ TAU system, which contains the executive, is the most tightly controlled. The BASIC software is controlled by engineering committee. In combination, these software tools simplify the development and validation of the application software.

Developed in-house, the μ TAU system has evolved into a complete real-time system for running flight simulations on the DEC VAX computers at SimLab.³ Hardware drivers, Ethernet protocols, and other system functions are transparent to the user thus allowing more time for developing the simulation. Some of the μ TAU system functions are to respond to the real-time clock, read and write to the I/O devices, call the aircraft model, and run CASPRE. Division of the VAX host computer frame is shown in Figure 3. Frame time varies depending

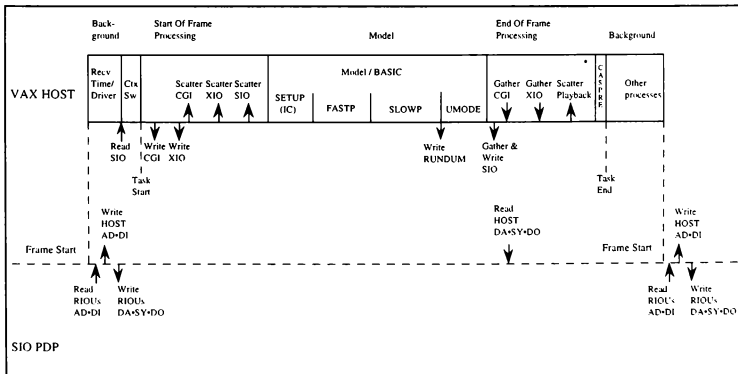


Figure 3 μ TAU Real-time System Timeline

on the complexity of the aircraft model and the number of I/O devices used. For the Space Shuttle landing and roll out model run at SimLab, the model took about 3.3 milliseconds and the other functions took about 7.6 milliseconds.

Additional features of the μ TAU system include the ability to execute a model non-real-time to facilitate debugging, the ability to record data to be displayed on the user's terminal through video strip charts, and the ability to plot all tabular data received for the model. Some simulations take advantage of a playback and record utility which allows data to be recorded and read back into a program or a visual target at a later time. Another feature of the system is the ability to spawn a DEC Control Language (DCL) subprocess to edit files while the CASPRE real-time process continues. A μ TAU facility to send output data to a disk, a terminal, or a printer has also been a valuable tool.

The CASPRE debugger contains many of the features present in the VAX debugger. Common features include the ability to set breakpoints and watchpoints, to examine and deposit into variables, to display source code and to read in text files by executing a command procedure. CASPRE, unlike the VAX debugger, is an on-line, real-time debugger. Other differences which help make CASPRE a powerful debugging tool are dumps of designated blocks of memory, searches through source code for occurrences of variables, and timing of the entire system or of individual routines.

CASPRE splits the screen window into several windows each displaying specific information. The output window displays the model output and the state of the executing model. CASPRE directives are shown in the command window and source code can be displayed in the source window. The monitor window can update up to twelve selected variables every three seconds. All of CASPRE's capabilities can be accessed during real-time operation allowing fast transitions from one flight condition to another.

Another tool is the BASIC library which contains routines that represent generic models which have been tested and validated. These routines can be transferred into the user's account and modified to reflect any additional features needed for a particular simulation. Some of the available routines are a kinematics (equations-of-motion) routine, a landing gear routine, a steady state wind routine, a turbulence generator routine, an actuator routine, and a filter routine that can simulate up to 20th order polynomial transfer functions.

Development

Development of a simulation begins with a pre-simulation conference involving the researchers

of the project and the staff at SimLab. The scope, purpose, and requirements of the study are agreed upon and a mathematical model is delivered. Options for the configuration of the separate simulation components are discussed and finalized. A team of engineers is assigned to the project and they stay with the project from development, through validation and integration, into operation and documentation. This team is responsible for ensuring that each component progresses toward completion, coding the mathematical model of the aircraft to be simulated, and validating the results using a variety of static and dynamic check-out cases.

After receiving their requirements, each of the hardware, graphics, and cab buildup groups determine the time required for the given task and a schedule is developed. The time from start of work to production in the VMS can vary. For example, simulations that are continuations of past efforts at SimLab may take three months while new simulations may take six months to a year to develop. The development time for the model may be shortened by using some of the generic routines that reside in the BASIC library. Reducing the development time allows a greater number of operational simulations per year.

The simulation engineer's primary job is to code the math model thus creating an accurate representation of the aircraft to be simulated. A computer model of the aircraft is coded using the structure developed in the BASIC software and is comprised of FORTRAN subroutines and tabular data. The μ TAU system has the capability of plotting the compiled tabular data for verification against the supplied data. The μ TAU linker takes the FORTRAN subroutines and function table data and builds an executable image. This image is then activated using the CASPRE feature of the μ TAU system.

All aircraft math model and BASIC library subroutines have an Initial Condition(IC) - Hold - Operate mode structure. During IC mode, initial condition values are deposited into variables each cycle allowing an easy setup to any initial condition within a flight envelope. In IC mode, the output of filters and all integrators are initialized to their steady state values. The simulation engineer can then select Operate (run) mode, which starts all state integration and activates all filters. The Hold mode freezes the model at the current time step. Interrogation of the status of the program through CASPRE can be done in any of the modes.

The μ TAU system and CASPRE allow the simulation engineer to select and send any desired variables to the strip charts. These assigned variables can be changed at any point during real time allowing easier debugging and trouble shooting of any problems.

During this development phase, extensive

checks are run to ensure model fidelity and continuity of the simulation. In order to run dynamic checks, inputs to the aircraft controls are created in DYNCHK, a BASIC routine. Some of the inputs transmitted to the model during the operate mode include steps, doublets, and frequency sweeps. Comparison of end-to-end static and dynamic checks with those independently generated by the researcher validates a model. Depending upon the complexity of the math model, stand-alone checks may be made of each module that comprise the simulation model. The complete end-to-end checks are performed again at the end of data production. This provides a baseline for future use of the simulation model. Some research objectives require modification of the math model during production. The end-to-end checks provide documentation of the changes made.

While the engineering team is preparing the aircraft model, cab development progresses in parallel. The cab structure is built around a pilot design eyepoint and the rest of the cockpit is developed around that point. Panels are built to contain the appropriate instruments, buttons, and switches. The aircraft controls, monitors, seats, and HUDs are mechanically aligned with the design eyepoint. The cab is cabled, wired, and prepared for integration with the aircraft model.

Integration

Software integration with cab hardware is normally accomplished during a two week period prior to the scheduled start of the ICAB fixed base or VMS data production session. Some projects are scheduled to remain in the ICAB lab for data production for further design and concept development in a fixed base environment. The majority are scheduled to move up to the VMS lab after the completion of cab integration. This two week period is referred to as the ICAB session and involves such tasks as cab checkout and lab setup.

McFadden control loaders are a central part of the cab hardware producing force gradient, damping, break out, hard stops, friction and trim positioning cues. These loader characteristics can be controlled by the host computer. Some of the control loader configurations available are a 2-axis control stick, a wheel and column, a collective and pedals. Force versus displacement is plotted for each control configuration and the force characteristics are verified against the specifications. Other cockpit controls include a 3-axis side arm controller and a throttle quadrant. As part of the integration process, all of the analog cockpit control displacements are readily scaled and biased using CASPRE.

Cab instruments are first checked for proper operation. Next, each instrument is calibrated using

the INSCAL routine from the BASIC library. A test value is sent through INSCAL to the instrument. If the instrument does not show this test value correctly, the scale and bias values are adjusted through a switch in the cab. INSCAL stores the adjusted scale and bias values and prints the new values to the engineer's terminal for update. Modification of the instrument driving routine may be needed to complete the integration process. In this fashion, the instruments can be checked out expeditiously, allowing the layout of the cab to be tailored to fit the needs of each simulation.

Sound cues are defined by the researcher for the particular vehicle being simulated as well as for the flight tasks to be performed during data production. Examples of some of the sound cues provided by the Wavetek generator and the Mirage Digital Sound Sampler are engine noise, rotor thrust, landing gear touchdown, wind noise, warning tones and weapon sound effects.

The operation of additional hardware, such as the Integrated Head And Display Sighting System (IHADSS) or the seat shaker, is checked during the ICAB session. Routines to drive these devices are available from the BASIC library.

Communication is established between the host computer and the attached processors which are used for graphics or auxiliary processing. This involves defining the data interface including data lists and transfer rates. The μ TAU real-time system allows the host to send a maximum of five different data packets. Several attached processors can be grouped to read the same data packet.

The four interchangeable cabs have up to four monitors for visual scene display, depending upon the cab design. An alignment procedure is followed to make the horizon continuous across all the monitors by adjusting the visual scene sent to these monitors. Once this has been accomplished, the HUD horizon is matched, for all bank angles, to the true horizon as displayed on the monitors. Both the HUD and window alignments are performed based on the pilot design eyepoint in the cab.

In addition to the CASPRE window, the Simulation Engineer Panel (SEP) and Project Engineer Panel (PEP) are used to control the operation of the simulation. These panels located next to the engineer's terminal provide a means of interacting with the model running in real-time. Each panel consists of thirty-two push-button switches and indicators configurable to the needs of the simulation. Eight of the switches and indicators are preassigned due to their common usage in all simulations.

The ICAB session is used to complete the setup of the data collection requirements for the simulation. A printout of summary variables and statistics is customized to the researcher's needs.

Digital data is recorded using RUNDUM, a software facility that collects real-time simulation data and stores it to a VAX hard disk. RUNDUM input files, which consist of lists of variables to be recorded and the recording rates, are created. Assignment files for the strip chart recorders, which provide analog data, are also created. These files are quickly and easily read in using the CASPRE window, allowing the task of data collection to run smoothly during the production phase.

Production

When scheduled for motion, the cab is moved from the ICAB area to the VMS on or before the first day of data production. Once the cab is secured on the motion system, a complete hardware checkout is done to ensure all systems are working after the move. The monitors in the cab, the HUD, the IHADSS, the seat shaker and all the instruments and controls are thoroughly checked.

The motion system is tuned during the first day of operation. This entails modifying the motion gains to optimize the motion for the type of vehicle and the tasks to be simulated. Ideally, the entire envelope of the motion system will be used with minimal triggering of the software or hardware limits.

Once the motion gains are determined, they will remain constant throughout the data production phase, unless the emphasis of the simulation is modified.

A simulation shift is eight hours per day for a five to six week period. Each morning, a motion check is performed to validate the motion system performance and is monitored from the lab. A picture of the VMS lab is presented in Figure 4.

Checkout of the real-time recording device ensues. RDUMP, a feature of μ TAU, displays the contents of a file created by RUNDUM and can convert it to various file formats. An option in RDUMP displays the statistics of all variables enabling the engineer to determine if the recording device is functioning properly.

Checkout of the cab is next. Control loaders and instruments are checked for accuracy. Meanwhile, the graphic systems are brought on-line with the appropriate displays. The simulation engineer then flies the aircraft model to verify that all systems are working.

The lab setup is checked to ensure that all monitors are displaying the correct video images. Video recorders are used to record out-the-cockpit scene with the HUD overlay and/or the glass cockpit displays. The strip chart recorders and the high speed



Figure 4 VMS Lab

printer, used to print summary variables and statistics for the run, are checked.

Setup of the production run follows. The specified environmental conditions are set through CASPRE. The setup may include setting wind direction and magnitude, control system parameters, aircraft configuration, a waypoint course or other program variables. These parameters establish the new initial condition for the run.

Once setup is complete, the pilot enters the cockpit and the cab is taken to the initial position to start the run. Communication among the pilot, motion operators and simulation engineers is kept on-line through an intercom system. The motion operators place the system in operate mode once the initial position and acceleration limits are within safety constraints. When the simulation is in the operate mode, control inputs are received from the pilot. At any time during the task, the pilot or the engineer can stop the motion system using a reset button and the program will go back to the initial conditions.

The duration of each run varies with the research project. Some runs take a couple of minutes, the time it takes for the space shuttle to descend from ten thousand feet to the ground, while others can take more than half-an-hour, the time for a UH-60 following the pathway-in-the-sky course. Variables are set quickly between runs, while the pilot is in the cab, minimizing turn-around time. Changes due to unforeseen circumstances or to pilot suggestions can be made to the code and an executable image can be built and put in operation within minutes. During the run, the performance of the vehicle and other aircraft parameters are checked through the strip chart recorders. At the end of every run, additional performance parameters are printed on the lab high speed printer.

Conclusion

The VMS is a flexible and dynamic research and development facility. The capability to provide a configurable environment while maintaining the fidelity of the research is of utmost importance. Assigning teams to each simulation allows concurrent development of different projects and gives each engineer the knowledge to make software changes efficiently. The high speed VAX computers at SimLab, combined with powerful software tools and the interchangeable cab system, enable the engineers to support a full schedule of simulations. With a new visual system, to be added later this year, the VMS will provide an even more realistic simulation environment. In the future, the VMS will continue to make an important contribution to the study of aircraft controls, handling qualities and guidance systems.

Acknowledgments

The authors would like to thank Burnett Lee and Jeannine Shirley for their assistance.

References

- 1 Danek, George, Vertical Motion Simulator Familiarization Guide Components and Systems, NASA TM-103923, To Be Published.
- 2 Cook, Anthony M., " Simulation World Moves up to V/STOL ", Aerospace America, November 1985.
- 3 μTAU Real-Time System User's Guide, Contract No. NAS 2-12859, NASA Ames Research Center, Moffett Field, CA 94087.

AN ASYNCHRONOUS SIMULATION METHODOLOGY FOR IMPROVING THE PERFORMANCE OF TRAINING SIMULATORS

W.H. Deiss, E.J. Fadden,* and R.M. Howe**

Applied Dynamics International,
Ann Arbor, Michigan 48108

Abstract

Partitioning a large simulation among processors in a distributed processor system so that each processor is used efficiently and the required data transfers between processors are minimized is very difficult. One approach is to assign tasks to processors on the basis of identifiable subsystems in the simulation. This paper describes a method that can be used for processor interfacing in general multiprocessor simulations with various hardware subsystems, including embedded controllers, with each processor and hardware subsystem operating at different frame rates that can be completely asynchronous with respect to one another. When data transfers between processors and I/O subsystems include time tags, extrapolation based on real-time integration algorithms can be used to reconstruct the data to provide accurate values at the required destination times. The use of a variable step integration method for real-time simulation is introduced. Variable-step real-time integration algorithms and associated extrapolation formulas permit automatic assignment of real-time integration step sizes, with an accompanying improvement in accuracy and computational efficiency. A scheme for asynchronous data transfers which provides more efficient utilization of data-bus bandwidth and major simplification in the programming of data transfers is described. All data transfer latencies are automatically compensated by the built-in extrapolation algorithms. Examples illustrate the concepts.

1. Introduction

Commercial off-the-shelf (COTS) single board computers (SBCs) that exploit the latest in microprocessor technology are available from a variety of sources, particularly in VMEbus format. It is now possible to assemble a very powerful, expandable, distributed processor system using a number of SBCs, where each SBC has one, two, or more microprocessor devices, 16 to 128 Mbytes of memory, and floating-point

performance in the range of 50 to 100 Specmarks or more. Assembling the necessary computer hardware resources is not where the challenge lies.

Training simulator manufacturers have long sought an easy solution to the difficult task of partitioning a massive training simulation model among several different processors to obtain an effective increase in compute power and hence a faster solution rate. Finding an easy solution to this partitioning task has been difficult to say the least.

The current popular approach to model partitioning requires the synchronization of each of the resulting tasks and thus of each processor. In order to ensure synchronization of all of the various tasks, this approach always results in some sizeable waste of time as the various processors invariably idle in a wait state looking for a system interrupt to signal the start of a new frame.

The totally synchronized approach to model partitioning has the appearance of making the overall programming job less formidable. However, developing a synchronization schedule for all the tasks in a large, complex model is not simple, particularly when various subsystems have disparate sampling rates and some activities have to be handled on an interrupt basis. A thorough knowledge of the model dynamics and the distributed processor system data rates is an absolute necessity with this approach. Once this information has been obtained, a time flow-chart of equation ordering must be carefully constructed. Even after all this has been accomplished, there is often a great deal of time spent in "tuning" the dynamics to meet even a minimum standard of operation. While the problem may be alleviated somewhat by the use of faster processors, the basic problem still does not go away.

This totally synchronized approach to the use of distributed processors is further complicated by the number of computer-based subsystems (i.e., visual, control loading, etc.) from different vendors that are used in a modern training simulator. These different subsystem manufacturers generally accommodate the simplest and most commonly used interfacing methods and protocols. This gives their products broad market appeal. However, the proprietary software packages used in these subsystems have generally been designed to minimize the amount of data transferred in each frame. Minimizing the

* Member, AIAA

** Associate Fellow, AIAA

size of the data transfer file does not necessarily simplify life for the system integrator.

Extrapolation techniques have been used in the past to interface a subsystem with a host where the frame rates of the host and the subsystem were not compatible. In most cases, the extrapolation techniques used have left a lot to be desired. The major problem has been reluctance on the part of the system integrator and/or the subsystem manufacturer to add the derivatives to the data transfer file needed to do an effective job of extrapolation. Consequently, these attempts to use simplified extrapolation techniques to reduce system latency effects have often resulted in jittery overshoot problems in the fringe areas of normal operation. These experiences have resulted in a reluctance to utilize extrapolation methods to any great extent.

On the other hand, the further fracturing of the model caused by partitioning it among processors using different communication bus protocols (e.g., MIL-STD-1553, ARINC-429, etc.) has made the utilization of extrapolation algorithms even more desirable, since the potential gain in delay recovery is even greater.

The task synchronization requirement imposes a severe burden on the simulator engineer and has a direct, adverse impact on simulator costs. One way to reduce the costs of model programming, system integration, and testing is to reduce or eliminate the need for task synchronization and let the various tasks interrelate in an asynchronous fashion.

This paper describes an asynchronous methodology for improving the performance of large simulation systems including training simulators. This methodology includes a form of real-time, variable-step integration of the system dynamics. Simple examples are used to illustrate the concepts discussed. The SIMsystem of Applied Dynamics International is briefly described. This multi-processor system will be used as the vehicle for validating the concepts using significant non-linear models. An overall multi-processor protocol is described, which is extremely easy to implement using the SIMsystem software tools.

1.1 Advantages for Training Simulators

There are a number of major advantages of utilizing the techniques proposed here in training simulators, including the following:

1. The simulator engineer is completely freed from the burdensome task of maintaining integer frame ratios between subsystem simulations. In fact, unless otherwise programmed, all simulation subsystems are free-running at their maximum frame rate, with variable frame-by-frame execution times automatically incorporated.

2. The time-consuming task of explicitly programming the real-time data transfers between processors and hardware subsystems, including I/O, is completely eliminated.

3. Changes and augmentations to the simulation, including the addition of more processors, are easily handled without the necessity of any significant reprogramming of the real-time data transfers.

4. All latency effects resulting from data-transfer delays are automatically compensated.

5. Any latencies in on-line hardware, such as CGI delays in visual displays, will be automatically compensated in the most effective possible way, namely, with an extrapolation formula which is identical with the integration algorithm used for the real-time simulation.

6. On-line calculation of local integration truncation errors and extrapolation errors provides instant feedback to the simulator engineer regarding the possible presence of unacceptable dynamic errors.

7. When the "freeze" command is executed in a simulation, all dynamic states will be automatically extrapolated to exactly the same freeze time, regardless of individual differences in subsystem frame rates.

8. Complex multi-processors, multi-hardware simulations can be interconnected by means of a graphic user interface (GUI), with all data flow and timing problems taken care of automatically.

In summary, it is believed that the techniques described in this paper reduce the problem of interfacing complex real-time digital multi-processor, multi-hardware simulations to the equivalent simplicity of interconnecting completely continuous systems, with the added bonus of on-line calculation of dynamic errors.

1.2 Overview of Basic Concepts

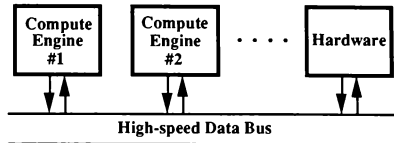
The rapid advances in microprocessor technology have heightened interest in methods for improving overall computational performance through the parallel operation of a number of separate processors. The difficulty of this approach has long been recognized as lying in the partitioning of the problem between processors in such a way that each processor is used efficiently and the required data transfers between processors are minimized. Development of efficient general-purpose compilers to perform this task has proven to be elusive. The problem is made even more difficult when it is necessary to perform the computation in real time, including cases where hardware-in-the-loop (HITL) is present, as in training simulators.

One approach to parallel computation is to assign tasks to individual processors directly on the basis of

identifiable subsystems in the overall simulation. This will, in general, tend to minimize the required number of data transfers between processors. This procedure is obvious and is not new. However, difficulties arise when the various subsystems making up the overall simulation have widely varying dynamic characteristics. For example, some subsystems may be slow and others fast, with each subsystem requiring different numerical integration frame rates for acceptable dynamic accuracy. When some of the subsystems are present in the form of hardware rather than computer simulations, the required input-output data rates for the hardware may again be different, especially in the case of digital control systems. In fact the subsystems may not even operate at a fixed frame rate. For example, this can be the case in some pulse-width modulated systems. Interfacing these many subsystems with different and perhaps variable frame rates without causing unacceptable dynamic errors can present an extremely difficult and challenging problem. As high technology systems employ more microprocessor control loops, these interfacing problems, both in the simulation and in the actual system being simulated, will continue to grow.

In this paper we consider the assignment of processors in a distributed processor system to identifiable subsystems in the overall simulation, as described above. The interfacing problem between processors is greatly simplified by letting each microprocessor or system run at its own frame rate, independent of the other frame rates. When a block of data is passed from one processor to another, it is accompanied by a data time-tag that identifies the discrete time represented by the data. This then permits the processor receiving the data to reconstruct through extrapolation the value of each variable at the time required for use in its own simulation algorithm or physical process. Figure 1 illustrates the architecture and a simple, first-order extrapolation formula, which in the figure is used to estimate the variable X at time $t = t_k$ with a first-order algorithm based on X at time $t = t_n$ and the estimated time derivative of X , as derived from a backward difference. Here it can be assumed that the data sequence $\{X_n\}$ is generated in one processor (compute engine) by a numerical integration algorithm with time-step size T , i.e., a frame rate given by $1/T$. Alternatively, the data sequence $\{X_n\}$ may originate in a hardware subsystem, in which case X_n is considered a real-time input to the overall computer simulation. It can be assumed that the data sequence $\{X_k\}$ is the required input to another subsystem simulation using an integration time-step size h . Or X_k may represent the required input to external displays or hardware in a training simulator. Note that the two step sizes, T and h , need not be equal. Nor do they need to be related by an integer ratio.

Although the integration and input/output step sizes are normally fixed in a real-time simulation, this is also not necessary for the multi-processor methodology presented here. Within the accuracy of the extrapolation algorithm, each data-point value is always converted to a representation at the point in time required by the subsystem receiving the data.



Bus data: each block of data consists of data words plus time tag, i.e., $X_n, Y_n, Z_n, \dots, t_n$

$$\text{Then } \hat{X}_k = X_n + \frac{X_n - X_{n-1}}{T} (t_k - t_n)$$

Figure 1. Multi-processor architecture.

The effectiveness of the procedure is best illustrated by specific examples. This is accomplished in the next two sections. Subsequent sections of the paper introduce the use of variable-step predictor integration methods and extrapolation algorithms, along with a description of an overall multi-processor data-transfer protocol, which is extremely easy to implement on the SIMsystem of Applied Dynamics International.

2. Example of a Multi-processor Simulation

Consider the flight-control system shown in Figure 2. It consists of an airframe and a pitch autopilot designed to produce an aircraft pitch angle θ in response to an input pitch-angle command θ_i . The dynamic behavior of the airframe is dominated by the short-period pitching mode, which in our example has an undamped natural frequency $\omega_{n_s} = 5$ rad/sec (≈ 0.8 Hz). The pitch control system produces an elevator command displacement, δ_{e_c} , equal to $K(\theta_i - \theta - C_d Q)$, where K is the autopilot gain, Q is the aircraft pitch rate, and C_d is the rate-constant. The elevator command displacement δ_{e_c} is converted to the actual elevator displacement δ_e by the elevator servo actuator system. The dynamic behavior of the pitch control system is dominated by the elevator actuator servo, which we will assume has an undamped natural frequency $\omega_{n_e} = 40$ rad/sec (≈ 6.4 Hz).

We will consider the pitch control system a fast subsystem to be simulated with an integration step size h and the airframe a slow subsystem to be simulated with an

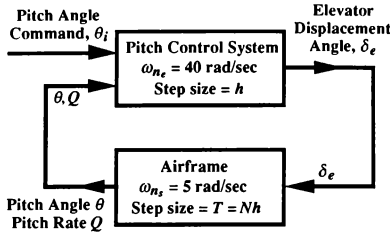


Figure 2. Flight control system.

integration step size $T=Nh$, where N represents the ratio of the fast frame rate to the slow frame rate. In multi-rate integration, the frame-rate ratio N is normally chosen to be an integer.¹ As a first example, we consider a multi-rate simulation with $h=0.01$ seconds for the pitch control system simulation and $T=0.05$ seconds for the airframe simulation. Thus, the frame-rate ratio $N=5$. The AB-2 integration algorithm is used in simulating both subsystems. As is evident in Figure 2, the pitch control system simulation requires inputs θ and Q from the airframe simulation. Since these will be generated with step size $T=0.05$ (frame-rate = 20 Hz) in the airframe simulation, they must each be converted to a data sequence with step size $h=0.01$ (frame-rate=100 Hz) to serve as inputs to the pitch control system simulation. In the case of the pitch rate Q this will be accomplished using the first-order extrapolation algorithm shown in Figure 1. In the case of the pitch angle θ , the following first-order extrapolation algorithm is used:

$$\hat{\theta}_k = \theta_n + Q_n(t_k - t_n) \quad (1)$$

Here the pitch-rate Q , which is equal to $d\theta/dt$, is used directly, instead of the backward difference $(\theta_n - \theta_{n-1})/T$, to represent the time-rate-of-change of θ in the first-order extrapolation formula. This provides improved extrapolation accuracy.² For $N=5$ the fast subsystem (i.e., the pitch control system) will require 5 inputs over the n th slow frame, inputs representing θ at $t_k = t_n, t_n + h, t_n + 2h, t_n + 3h$, and $t_n + 4h$. It follows that the fast data sequence over the n th frame is generated using the following formulas:

$$\begin{aligned} \hat{\theta}_n &= \theta_n, \quad \hat{\theta}_{n+2} = \theta_n + .2Q_nT, \quad \hat{\theta}_{n+4} = \theta_n + .4Q_nT, \\ \hat{\theta}_{n+6} &= \theta_n + .6Q_nT, \quad \hat{\theta}_{n+8} = \theta_n + .8Q_nT. \end{aligned}$$

In the case of the pitch-rate Q the time derivative \dot{Q} is not available at the start of the n th frame and, as noted above, the backward-difference extrapolation formula shown in Figure 1 is used to generate the required multi-rate data

sequence. In this case the extrapolation algorithm is given by

$$\hat{Q}_k = Q_n + \frac{Q_n - Q_{n-1}}{T}(t_k - t_n) \quad (2)$$

Over the n th frame for $N=5$ the formulas become

$$\begin{aligned} \hat{Q}_n &= Q_n, \quad \hat{Q}_{n+2} = Q_n + .2(Q_n - Q_{n-1}), \\ \hat{Q}_{n+4} &= Q_n + .4(Q_n - Q_{n-1}), \quad \hat{Q}_{n+6} = Q_n + .6(Q_n - Q_{n-1}), \\ \hat{Q}_{n+8} &= Q_n + .8(Q_n - Q_{n-1}). \end{aligned}$$

In the simulation of the pitch control system, both the elevator displacement δ_e and the displacement rate $\dot{\delta}_e$ are state variables. This means that the displacement $\delta_{e,n}$, as needed at time t_n for the n th frame input to the airframe simulation, can be obtained by extrapolation from $\delta_{e,k}$ and $\dot{\delta}_{e,k}$ using the formula

$$\hat{\delta}_{e,n} = \delta_{e,k} + \dot{\delta}_{e,k}(t_n - t_k) \quad (3)$$

For multi-rate integration with $N=5$, as initially considered here, $t_n = t_k$ at the beginning of the n th slow integration frame, so that extrapolation is not necessary. However, when we consider multi-rate simulation with non-integer values of the frame-rate N , Eq. (2) will need to be employed, since in that case t_k will, in general, not equal t_n at the start of the n th frame.

For typical autopilot parameters, the aircraft response θ to an acceleration-limited step input θ_i is shown in Figure 3, both for the ideal continuous case and for the numerical simulation using multi-rate AB-2 integration, as described above for integration step sizes $T=0.05$ and $h=0.01$. Note that the simulation output data points θ_n fall close to but not exactly on the ideal response curve. By using the acceleration-limited step input instead of the ideal step, we avoid the discontinuity in displacement and velocity associated with a true step input. This in turn provides a more suitable and realistic input when using predictor integration methods in the simulation. Otherwise, the dynamic errors in the simulation tend to be dominated by those resulting from the discontinuities in the input rather than integration truncation errors associated with the overall simulation. The response of the flight control system in Figure 3 to the acceleration-limited step input is very nearly the same as the response to an ideal step input applied at $t=0.2$ seconds, which is halfway through the 0.4-second transition time of the acceleration-limited step. The ideal pitch angle response θ shown in Figure 3 was obtained using RK-4 integration with $h=T=0.01$ sec. With this very small step size, the errors due to integration truncation errors are less than 10^{-6} , and the solution can be used as a reference for

determining the dynamic errors when using larger step sizes.

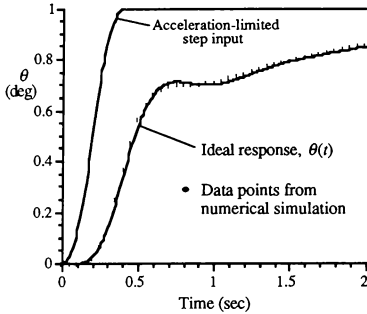


Figure 3. Pitch angle response to acceleration-limited step input.

2.1. Simulation with Non-integer Frame-rate Ratios

We next consider multi-rate simulations for $T = 0.055236$ and $T = 0.044764$, i.e., $T = 0.05(1 \pm \pi/30)$ instead of $T = 0.050000$, with h still equal to 0.01. The frame ratios N for these two new cases are equal to $5 \pm \pi/6 = 5.5236$ and 4.4764, respectively, both irrational numbers. Figure 4 shows the simulation results, along with the previous solution for $N = 5$. In all three simulations, Eqs. (1) and (2) are used as needed to generate the multi-rate inputs θ and Q to the fast pitch control system from the slow airframe system, and Eq. (3) is used as required to obtain the input δ_e to the slow airframe system from the fast pitch control system. Little difference can be discerned between the simulated outputs in Figure 4, even though two of the three simulations are running with completely asynchronous frame-rate ratios.

In Figure 5, the errors in simulated output for all three cases are shown, where the errors represent the difference between the results of Figure 4 and the reference solution shown earlier in Figure 3 for the acceleration-limited step input. The differences in output caused by using the three different step sizes T in the simulation of the airframe are now readily apparent. In general, the differences are less in magnitude than the errors themselves, and therefore do not contribute significant additional error to the overall simulation. In fact, the difference in errors between the various solutions is mostly due to the integration

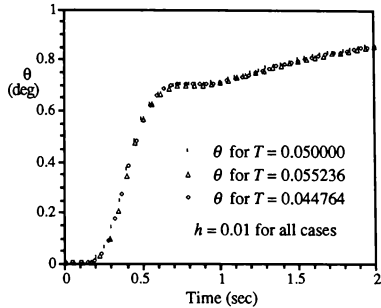


Figure 4. Simulated output θ for three different slow subsystem frame rates.

truncation errors resulting from the different step sizes T used in each case for the airframe simulation. Since the AB-2 algorithm used for both slow and fast subsystem integrations is second order, the dynamic errors in the airframe simulation should be proportional to T^2 , the square of the step size. This in turn suggests that the simulation for $T = 0.055236$ should exhibit errors which are $(.055236/.05)^2 = 1.22$ times the errors when $T = 0.05$. Similarly, for $T = 0.044764$, the errors should be $(.044764/.05)^2 = 0.80$ times the errors when $T = 0.05$. Comparison of the errors in Figure 6 for all three step sizes shows that the above error ratios hold, at least approximately. Thus we conclude that the additional error contributions of the extrapolation algorithms used to interface the fast and slow subsystem simulations are indeed quite negligible.

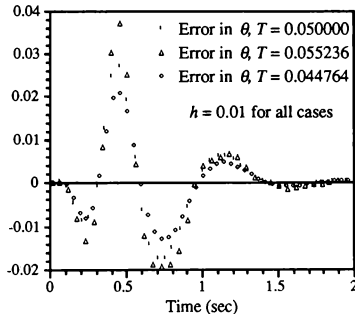


Figure 5. Simulated output error for three different slow subsystem frame rates.

Close inspection of the data points in Figure 5 also shows the different step times associated with each of the three step sizes T used in the airframe simulation. For $N=5$, i.e., $T=0.05$, the dimensionless extrapolation interval $a = (t_k - t_n)/T$ takes on the 5 values 0, 0.2, 0.4, 0.6, and 0.8 over each slow frame of duration 0.05. On the other hand, for $N=5.5236$ ($T=0.055236$) the number of values taken on by a each slow frame can be either 5 or 6. For $N=4.4764$ ($T=0.044764$) a takes on either 4 or 5 values per slow frame. In both cases, the values of a range between 0 and 1 and never repeat.

The frame-rate ratio N can also be changed by varying the fast subsystem time step h from its nominal value of 0.01 while maintaining the slow subsystem time step at $T=0.05$. When this is done using the above ratios of $N=5.5236$ and $N=4.4764$ (this corresponds to letting $h=0.011047$ and 0.008953 , respectively), the resulting simulation output errors are affected even less than the effects shown in Figure 5.³ See also the comment following Figure 6.

The simulation results presented in Figures 4 and 5 have not taken into account the effects of data-transfer delays between the fast and slow subsystem simulations, as in a multi-processor implementation. Later we will describe an asynchronous data-transfer protocol which will minimize these delays. If the data bus transfers between processors in Figure 1 can only be repeated at a frame rate equal to the frame rate of the fast subsystem, which is an extremely pessimistic assumption, then the average data transfer delay will be $h/2$, where h is the integration step size of the fast subsystem. As a specific example, we consider the case where the step size for the slow subsystem simulation is kept at its nominal value, $T=0.05$ seconds, while the fast subsystem step size is changed from its nominal value of 0.01 seconds to $h=0.008953$. Figure 6 shows the resulting errors in simulation output both with and without an additional data-transfer time delay of $h/2$, or 0.004476 seconds. Clearly the additional delay makes no significant contribution to the overall error. The reason, of course, is that the extrapolation algorithms automatically correct the transferred data for the delays. The effect of the delay on the extrapolation can be seen by examining the time history of the multi-rate dimension-less extrapolation interval, $a = (t_k - t_n)/T$, both with and without the 0.004476 second data transfer delay. When the data transfer delay is present, a ranges between 0.1117 and 1.1117. Without delay, a ranges between 0 and 1, as noted earlier.

Careful comparison of the results in Figure 6 for the case of no data-transfer delay, where $T=0.05$ and $h=0.008953$, with the results in Figure 5 when $T=0.05$ and $h=0.01$ shows that there is very little difference between the corresponding output errors. We conclude that

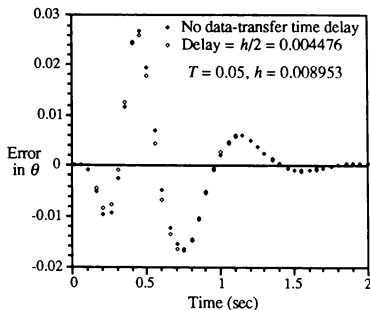


Figure 6. Effect of data transfer time delay.

changing the step size h of the fast subsystem simulation from 0.01 to 0.008953 seconds is almost completely compensated by the extrapolation algorithms of Eqs. (1), (2) and (3).

3. A Mixed Continuous and Digital System

As a second and more difficult example of the use of asynchronous simulation, we consider the flight control system when the controller is digital. In this case we must contend not only with the simulation of both continuous and discrete systems, but also with the simulation of the interface between the discrete and continuous system simulations. Figure 7 shows the block diagram, which again includes the continuous airframe system (simulation step size = T) and the continuous control-surface actuator (simulation step size = h), along with a digital control system (sample-period = T_{sam}) and a zero-order extrapolator, which converts the digital controller output data sequence $\{\delta_{ik}\}$ to the continuous input $\delta_i(t)$ for the actuator. As before, the dynamic behavior of the airframe is dominated by the 5 rad/sec undamped natural frequency, ω_{ns} , of the short-period pitching motion. Also, the control surface actuator is once more represented by a second-order system with undamped natural frequency, ω_{ne} , equal to 40 rad/sec. The following control law is assumed for the digital controller:

$$\delta_{ik+1} = K [\theta_{ik} - \theta_k - (C_d/T_{sam})(\theta_k - \theta_{k-1})] \quad (4)$$

Here δ_{ik} represents the digital controller output at the k th frame, K is the controller gain constant, θ_{ik} is the input

pitch angle, and θ_k is the airframe output pitch angle at the k th frame, as generated by the A to D converter with sample-period T_{sam} . Also, $(\theta_k - \theta_{k-1})/T_{sam}$ is the estimated pitch rate, as obtained from a backward difference approximation, and C_d is the effective rate constant.

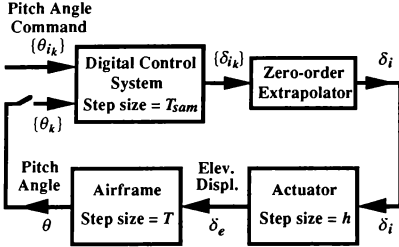


Figure 7. Digital flight control system.

For illustrative purposes we select a digital controller frame rate of 100 Hz (i.e., $T_{sam} = 0.01$ sec). We also employ the same values of gain constant K and rate constant C_d used previously for the flight control example in Section 2. For the acceleration-limited step input of Figure 3, the resulting pitch angle response turns out to be nearly identical with the response shown in Figure 3. We therefore conclude that the digital controller frame rate of 100 Hz is fast enough to provide about the same closed-loop performance as the continuous controller.

In addition to the digital-controller sample period $T_{sam} = 0.01$ sec, we choose a nominal integration step size of $T = 0.02$ sec for the airframe simulation and $h = 0.005$ sec for the actuator simulation. Thus the ratio between actuator and airframe integration frame rates in this case is given nominally by $N = 4$. Since there are two actuator integration steps per digital controller step, i.e., $T_{sam}/h = 0.01/0.005 = 2$, there is no problem in synchronizing the output samples of the zero-order extrapolator with the integration step times for the actuator simulation. In all cases, the sampled input θ_k to the digital controller is obtained using quadratic extrapolation based on θ_n , θ_{n-1} , and $\dot{\theta}_n$.² The quadratic extrapolation is necessary in order to obtain sufficient accuracy in the backward difference $(\theta_k - \theta_{k-1})$ in Eq. (4). The first-order extrapolation of Eq. (3) is used to generate the required input values δ_{e_j} and δ_{e_j} for the airframe simulation from the outputs δ_{e_j} and δ_{e_j} of the actuator simulation. When the errors in simulated airframe output for the nominal step sizes given above, including an airframe simulation step size of $T = 0.02$ sec, are compared with the

simulation errors for $T = .02(1 \pm \pi/30) = 0.022094$ and 0.017096 sec, results very similar to those shown earlier in Figure 5 are obtained, with the exception that the errors are all reduced by $(0.02/0.05)^2$ because of the smaller step size T used for the airframe simulation. Thus the extrapolation formulas used to compensate for the non-commensurate frame ratios are again quite effective, in that they produce incremental errors which are small compared with the basic errors associated with the finite integration step sizes used in the simulation. As before, most of the error differences for the three cases, $T = 0.02$, 0.022094 , and 0.017096 , can be explained by the second-order truncation error associated with each different step size T . It should be noted that the ideal reference solution against which the above simulation results are compared to determine the simulation errors must be recomputed for the sampled-data case here using RK-4 integration with a sufficiently small step size to ensure completely negligible numerical errors.

We next consider the effect of varying the actuator simulation step size h from its nominal value of 0.005 sec. In this case, the ratio of the digital-controller step size, $T_{sam} = 0.01$, to the actuator step size h will no longer be 2:1. This in turn means that sampling the extrapolator output δ_i to obtain the actuator input δ_j at the j th frame can result in a substantial error in the actuator simulation. The problem is illustrated in Figure 8, which shows the controller sample times t_k , the controller output samples δ_{i_j} , the corresponding extrapolator output $\delta_i(t)$, and the actuator frame times t_j for the case where $T_{sam} = 0.01$ and $h = 0.00544$ sec. Note that the extrapolator output sample δ_{i_j} at the time t_j is not representative of the extrapolator output over the entire j th step when the extrapolator output jumps to a new value before the step is completed. The problem is solved by actually computing the average extrapolator output over each actuator simulation time step using the following simple formula:

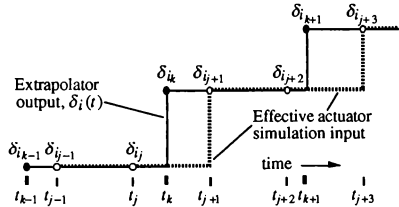


Figure 8. Extrapolator output and effective extrapolator input for a non-integer frame ratio.

$$\bar{\delta}_{ij,j+1} = \delta_{ij}(t_k - t_j)/h + \delta_{ij+1}(t_{j+1} - t_k)/h \quad (5)$$

The average extrapolator output over the j th actuator time step, $\bar{\delta}_{ij,j+1}$, then becomes the input to a modified AB-2 integration algorithm in the actuator simulation.⁴ Using this procedure, we obtain the results in Figure 9, where the error in simulated output is shown for three different actuator simulation step sizes, $h = 0.005000$, 0.0055236 , and 0.0044764 . Note that the change in step size h has very little effect on the simulation, which indicates that the use of Eq. (5) to compensate for the non-integer ratio T_{sam}/h that occurs when $h = 0.0055236$ and 0.0044764 is indeed quite effective.

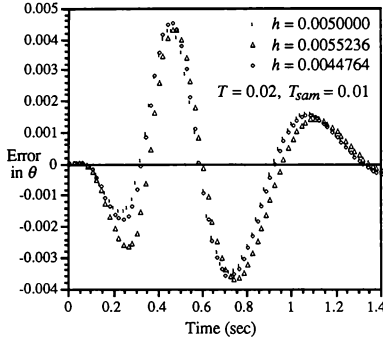


Figure 9. Effect of actuator step time h on the digital flight control system simulation.

4. Real-time, Variable-step Integration

In all of the examples considered thus far in the paper, we have assumed that each subsystem simulation utilizes a fixed integration step size, even though we have allowed non-commensurate integration frame-rate ratios. In fact, it has been traditional to use a fixed time step for numerical integration in the real-time simulation of dynamic systems. The overriding reason for not using a variable step size is the possibility that the mathematical step size can become smaller than the computer execution time for the calculations involved in a given integration frame. This in turn means that the simulation output at the end of that step will fall behind real time. Another reason for choosing a fixed time step in real-time simulation is the compatibility with fixed sample rates when dealing with real-time inputs and outputs. On the other hand, when conditional statements are present in the program that is

executed every integration step, the frame execution time will not be constant. Also, the refresh operation associated with dynamic random-access memory will, in general, cause small variations in the execution time for each integration frame, as will the utilization of cache memory, depending on the frequency of cache "hits." The mathematical step size must of necessity be set equal to the maximum expected value of the frame execution time, which often may not even be known in a complex simulation, in order to ensure the availability of simulation outputs in real time.

With the time tagging and extrapolation techniques described in this paper, many of the reasons listed above for always using fixed integration step sizes in real-time simulations are obviated. In this section we go even further by suggesting that variable-step real-time integration may be introduced purposely to improve computational efficiency when variable integration-frame execution times are present in a simulation. In particular, it is possible for a real-time computer simulation to be run with the mathematical step size for each successive integration step set equal to the measured execution time for the previous step. With this procedure, the simulation is always able to keep up with real time, at least to within a fraction of the integration step size. The procedure also permits the real-time integration step size to be set automatically by the software, without user intervention. On-line error estimates can be employed to alert the user if the simulation errors ever become excessively large. The use of variable-step integration with extrapolation and/or interpolation to compensate real-time inputs and outputs for lack of synchronization with the variable integration frame rates, as introduced in this paper, is especially attractive in large, multi-processor simulations, including hardware-in-the-loop (HITL).

First let us write the formula for second-order, variable step predictor integration. We let the integration step associated with the n th integration frame be denoted by h_n . From the appropriate Taylor series expansions it is easy to show that the formula for integrating the state equation $\dot{X} = F$ based on F_n and F_{n-1} is given by

$$X_{n+1} = X_n + h_n \left[F_n + \frac{F_n - F_{n-1}}{h_{n-1}} \cdot \frac{h_n}{2} \right] \quad (6)$$

For $h_n = h_{n-1} = h$, Eq. (6) reduces to the familiar AB-2 formula, $X_{n+1} = X_n + h[(3/2)F_n - (1/2)F_{n-1}]$. When we recognize that $.5h_n/h_{n-1}$ needs to be computed only once each frame, regardless of the number of state variables, it is clear that Eq. (6) requires 2 multiplications and 3 additions per state variable, compared with the 2 multiplications and 2 additions needed for conventional AB-2 integration. As a simple example, we apply the

variable-step algorithm of Eq. (6) to the simulation of a second-order linear system with $\omega_n = 1$ and $\zeta = 0.25$. We choose a nominal integration step size of $h = 0.2$, with the actual step size $h_n = 0.2[1 + 0.2\sin(4n)]$. This results in the time-varying step size shown in Figure 10, which oscillates about 0.2 within the interval 0.16 to 0.24 and could be representative of the variable frame execution time in an actual simulation.

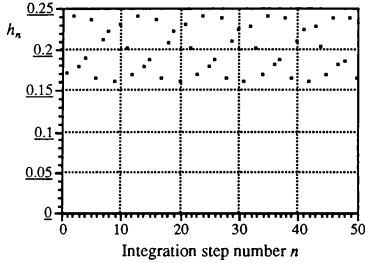


Figure 10. Integration step size h_n versus frame number n .

Using the variable-step predictor integration algorithm of Eq. (6) with the variable step size h_n given in Figure 10, we simulate the response of the second-order system to an acceleration-limited unit step input. The resulting output error is shown in Figure 11. Also shown in Figure 11 is the error when using a fixed step size of $h = 0.2$ (the mean value of the variable step size) and $h = 0.24$ (the maximum value of the variable step size). Note that the errors for the variable step case are essentially the same as those for the fixed step when the fixed step is equal to the mean, $h = 0.2$, of the variable step. On the other hand, if the fixed step is chosen to accommodate the maximum variable step size, $h = 0.24$ and the errors are significantly larger, i.e., by the ratio $(0.24/0.2)^2$.

In considering the results in Figure 11, it should be noted that not all of the variable-step output data points will be available in real time. This is because the mathematical step size is always one step behind the actual measured execution time for each step. Furthermore, the output data sequence in a real-time simulation is often required at equally spaced time intervals. For these reasons it is important to consider the

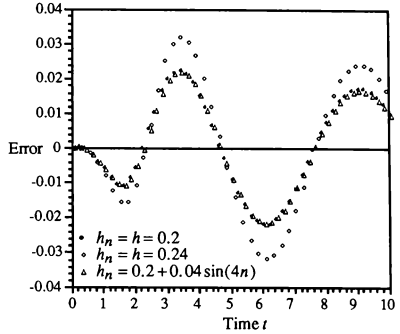


Figure 11. Simulation output error for both fixed and variable step sizes h_n .

reconstruction of a fixed-rate, real-time output data sequence X_k with step size h_k using extrapolation from the data points X_n . This can be accomplished using linear extrapolation based on Eqs. (1) or (2), by using quadratic extrapolation, or any other extrapolation formula. The most effective technique is to use extrapolation based on the same integration algorithm employed in the variable-step simulation, i.e., the predictor formula given in Eq. (6) for the example considered here. When Eq. (6) is used for extrapolation, the step h_n is simply replaced by the extrapolation interval $t_k - t_n$. Thus the formula becomes

$$\hat{X}_k = X_n + (t_k - t_n) \left[\dot{X}_n + \frac{\ddot{X}_n - \ddot{X}_{n-1}}{h_{n-1}} \cdot \frac{(t_k - t_n)}{2} \right]. \quad (7)$$

For the case where the fixed step h_k is equal to the mean, 0.2, of the variable step h_n , Figure 12 shows the errors in the fixed-step extrapolated output data sequence \hat{X}_k , along with the errors in the original, variable-step output X_n . Note that the incremental errors caused by extrapolation are essentially negligible. In fact, when the extrapolation interval $t_k - t_n$ is exactly equal to the size of the next integration step, the extrapolated result will by definition agree exactly with the integrated result, since the same formula is used for both calculations. This suggests that we would have achieved even better results in our asynchronous, multi-rate example earlier in the paper had we used Eq. (7) for extrapolation instead of the linear and quadratic formulas.

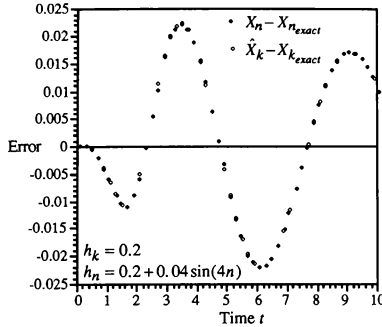


Figure 12. Errors in fixed-step data points \hat{X}_k generated from variable-step data points \hat{X}_n .

When using variable-step real time simulation with the step size for each frame set equal to the measured execution time of the previous frame, it is possible that the step size for one particular frame may become much larger than the mean step size. This in turn might cause an unacceptable error. For this reason it may be desirable to calculate an on-line estimate of the error resulting from each integration step. Indeed, such an estimate is a basis for selecting the step size itself in non-real-time, variable-step integration methods. In the case of variable-step, second-order predictor integration it can be shown that the local integration truncation error for the n th step is approximately equal to $-h_n^2(2h_n + 3h_{n-1})\ddot{F}_n/12$. This leads directly to the following numerical approximation for the local truncation error associated with the integration formula of Eq. (6):⁵

$$\text{LTE}_{n+1} \equiv -\frac{h_n^2\left(\frac{h_n}{3} + \frac{h_{n-1}}{2}\right)}{h_n + h_{n-1}} \left[\frac{F_{n+1} - F_n}{h_n} - \frac{F_n - F_{n-1}}{h_{n-1}} \right]. \quad (8)$$

For $h_n = h_{n-1} = h$, Eq. (8) reduces to the AB-2 formula for the local truncation error, $-5h^3\ddot{F}_n/12$.

It seems clear that an on-line numerical calculation of local truncation error can be relied upon to flag an unacceptable error resulting from too large a mathematical step size. When this occurs, the only recourse is to utilize a faster real-time computer or, in the case of a large simulation, to partition the simulation among several computers. In the latter case, the techniques described in this paper should prove highly effective. Note that Eq. (8) can also be used for on-line estimation of extrapolation errors as well.

In addition to the second-order predictor method considered here, variable-step formulas for higher order algorithms, such as third-order predictor integration, can be derived, as well as formulas for on-line calculation of the local truncation error.⁶

5. The Applied Dynamics SIMsystem

The SIMsystem is a family of products from Applied Dynamics International designed for real-time simulation applications that involve a tie to external hardware such as in a training simulator. A distributed processor version of ADI's SIMsystem is used to put the examples in this paper in a practical framework. Of particular interest in this paper are the AD Real-Time Station (AD RTS) and COSIM. A detailed overview of these components of the SIMsystem is contained in the paper "Improving the Performance of Flight Simulators via Smart I/O Interface Systems."⁷ Key characteristics of the AD RTS and COSIM, including pertinent additions that have been made since this paper was written, are reviewed briefly in this section.

5.1 The AD RTS

The AD RTS is a VMEbus-based, local area network (LAN) resource accessible from user workstations operating under UNIX or VMS. The AD RTS includes computer facilities, an intelligent I/O capability, optional ties to other bus systems (e.g., DR11, HSD, IEEE-488, MIL-STD-1553, ARINC-429, etc.), and VMEbus communications/control mechanisms.

Simulation compute power can be incorporated in the SIMsystem in a variety of ways. An external computer can be connected to the AD RTS via point-to-point (i.e., not part of a LAN) Ethernet or a standard interface such as DR11 or HSD. Single board computers (SBCs) incorporated in the AD RTS provide another means of adding compute power. ADI refers to SBCs used in this way as "compute engines" or CEs. Two types of CEs currently supported in the AD RTS are:

- The Motorola MC68040-based MVME167, denoted as a CE-1; and
- Two Motorola MC88110-based SBCs, Motorola's MVME397 and Tadpole Technology's TP810V, which are denoted as CE-2s.

In addition to these possibilities, ADI and the Computer Systems Division of Harris Corporation have entered into a technical and marketing agreement to jointly develop a product which combines the Harris Night Hawk with the SIMsystem in a single package. In addition to FORTRAN and C, the Night Hawk also provides Ada as a programming language.

The AD RTS contains two other computer resources that play special roles. One is the VMEbus Interact Manager, or VIM. The other is the Communication Processor, or COP. The VIM is a Motorola MC68030-based SBC with a 100 Mbyte disk and an Ethernet port that plugs into the VMEbus of the AD RTS and connects the AD RTS to the user's workstation via an Ethernet LAN. It provides the interface for all user interaction with the various facilities of the AD RTS. The VIM:

- Runs UNIX and manages all aspects of the connection with the user's workstation;
- Loads and starts other AD RTS system components and supports operations such as debugging and running diagnostics; and
- Handles all user interactive requests to the AD RTS at run-time and other run-time-related activities such as servicing data transfers for logging and graphing purposes that must be performed in a timely fashion, but are not highly time critical.

Since the VIM functions in a UNIX setting, it acts as a slave only on the VMEbus during run time (except for certain very limited activities such as halting the simulation) and does not play any role that would impact the proper real-time behavior of the simulation.

The COP is the vehicle for tightly controlling the time critical, run-time VMEbus traffic in the AD RTS. The COP uses an Advanced Micro Devices Am29000 as its controller and has special architectural features to facilitate data gather/scatter operations on the VMEbus. While the COP has a variety of capabilities, the one of prime interest in this paper is its control of all time-critical data communications. This communications control is handled via data gather/scatter operations. The program for the COP is generated by COSIM.

5.2 COSIM

COSIM is a distributed processor simulation executive and interactive run-time communication/control package that consists of two parts, namely, the COSIM language and COSIM Interact.

The COSIM language facilitates implementation of the asynchronous simulation methodology in a distributed computer environment by allowing the user to:

- Specify the distribution of programming tasks among the various parallel computational resources;
- Define any scheduling structure required, including any synchronization that may be needed;
- Program the communications sections of each of the parallel operations; and

- Support the various I/O and digital interface capabilities of the AD RTS through the use of an extensive run-time library.

From its founding in 1957 until the mid-1970s, ADI designed, manufactured, and sold analog and hybrid computers for simulation. In 1976, ADI began the development of the AD 10, which evolved over several years into a digital computer replacement for the analog and hybrid computers. Most of the customers for the AD 10 were people who had been analog or hybrid computer users. These people were very much accustomed to the "hands on" way of operating the analog and hybrid computers and expected the same kind interactive user-friendliness with the AD 10. Thus, ADI began developing its interactive approach to run-time software for a digital simulation system. COSIM Interact represents the cumulative experience of more than fifteen years of providing this type of capability for ADI users.

COSIM Interact provides the user with complete run-time control of the AD RTS from the user's workstation. This interactive package allows the user to monitor and control all aspects of the simulation such as:

- Loading, saving, and restoring simulations;
- Controlling all processors;
- Monitoring variables in a non-intrusive fashion;
- Capturing data for data logging and/or graphical display; and
- Symbolic debugging, including breakpoint management, single stepping, and variable manipulation, in a multi-language (i.e., ADSIM,[†] FORTRAN, or C), multi-processor environment.

COSIM Interact is fully integrated with the other components of the SIMsystem including the (optional) plotting package, SIMplot. Users can write session scripts, customize and extend the Interact command language and graphical user interface, and use the comprehensive real-time and off-line graphics facilities of SIMplot.

COSIM includes a Data Acquisition System (DAS) which underlies some of the features of COSIM Interact. In particular, DAS:

- Supports the retrieval of data to the user's workstation over the LAN during a real-time simulation; and
- Permits run-time, interactive selection of ADSIM, COSIM, FORTRAN, or C variables to be acquired and does not require that these variables be hard coded.

Data obtained via DAS can be displayed or analyzed during the course of a run to provide valuable information

[†] ADSIM is ADI's continuous system simulation language.

as opposed to having to wait for the completion of the run to obtain this information.

6. Multi-processor. Data-transfer Protocol

In this section, we consider a way of implementing the ideas discussed in the earlier sections on the ADI SIMsystem. Having introduced variable-step, real-time integration, which in turn led to the concept of data extrapolation based on the very same variable-step integration formulas, we now return to considering the overall asynchronous multi-processor methodology described in the first four sections of the paper. In particular, we demonstrated in Figure 6 that the extrapolation algorithms are extremely effective in compensating for time delays in data transfers between processors and hardware subsystems, including I/O modules. It is nevertheless important to arrange the data transfers such that delays are, in general, minimized and that data within any transfer is accurate. These goals are achieved with the following protocol:

1. For any given processor or hardware system the block of all data to be transferred to other processors or hardware systems is loaded into an output data buffer at the end of each local frame. When the time-derivative of the output data is also available, which is very often the case, this too is included in the data block. The data word representing the time tag (step size multiplied by the frame number) is also loaded into the output buffer. Finally, when the set of output data, including the time tag, has been loaded into the output data buffer, a data-ready flag is set in the output data buffer.

2. The COP in the AD RTS is programmed via COSIM to interrogate the output data buffer (ODB) of each processor or hardware subsystem in a specified order. The COP first reads the data-ready flag for each ODB and determines whether it has been set since the last interrogation. If it has not, the data in the ODB is assumed to be unchanged since the previous interrogation, and the COP moves immediately to the ODB for the next processor or hardware subsystem. If the data-ready flag for the given ODB has been set, then the COP proceeds to read the contents of this ODB, clears its data-ready flag, and distributes the output data, including the associated time tag, to the appropriate destination processors or hardware subsystems. Note that this procedure ensures that all data in a given ODB represents the same point in time, since the data-ready flag is the last thing to be updated in the ODB. Thus, if a given ODB is in the process of being updated when the COP interrogation occurs, the data-ready flag will not have been set and the COP will not read the data in this ODB. If the data-ready flag has not been cleared when a processor or hardware

subsystem is ready to load new data into its ODB, one of several possible alternatives is to put that processor or subsystem into a wait state until its ODB has been read and the data-ready flag has been cleared. Since such delays are undesirable, this requires that the maximum time between COP interrogations of a given processor be less than the frame time of that processor. This requirement can easily be met with a sufficiently fast bus system such as that in the AD RTS, especially if multi-rate COP interrogation is utilized for processors or hardware systems with the shortest frame times.

3. The COP interrogation and data transfers for processors and hardware systems is completely free-running. The ODB of every processor and hardware subsystem is interrogated in the specified order and data transfers to the specified destinations are only made when the data-ready flag for an ODB indicates a change in data since the last interrogation. This procedure ensures an overall maximum rate of data transfers. Since the accompanying time tag permits reconstruction of each data value so that it is correct for the required time at the destination, all delays in data transfers due to the asynchronous nature of the data bus transfer protocol are completely compensated, at least to within the accuracy of the extrapolation algorithms. The example simulations presented in Sections 2 and 3 have demonstrated the effectiveness of the extrapolation algorithms. In the case of processors or hardware systems with relatively short frame times, it may be desirable or even necessary to schedule multi-rate interrogations, as noted above.

Since each processor in the multi-processor scheme described in this paper can be free-running, it is important to ensure that any one processor does not fall behind or get ahead of the other processors by more than a fraction of its frame time. In a real-time simulation, this is usually handled by utilizing a mathematical step size (frame time) which is at least as large as the processor execution time for each frame. Computation of the next frame in that processor is then not started until the beginning of the next frame in real time. As an alternative to this, we can use the variable-step real-time integration described in Section 4, with the step size for each frame set equal to the measured execution time of the previous frame. This ensures maximum utilization of each processor and guarantees that all processor outputs occur in real time to within a fraction of their respective frame times. Again, any output variations from real time are very accurately compensated by the extrapolation algorithms. It is not even necessary to synchronize the internal clock of each processor with a master, real-time clock, providing the individual processor clock times remain accurate to within a fraction of a frame time over the total duration of the simulation. Note that this same

scheme can be used in non-real-time, multi-processor applications, where the entire simulation runs at a rate which is a specified speedup or slowdown factor with respect to real time.

7. Conclusions

The concept of completely asynchronous operation of multi-processors and hardware subsystems in a real-time simulation has been demonstrated. When data transfers between processors and I/O subsystems include time tags, extrapolation based on real-time integration algorithms can be used to reconstruct the data to provide accurate values at the required destination times. Simulation examples, including a combined continuous/discrete system, have shown that each processor can be operated at different, non-commensurate frame rates with no degradation in accuracy. Variable-step real-time integration algorithms and associated extrapolation formulas permit automatic assignment of real-time integration step sizes, with an accompanying improvement in accuracy and computational efficiency. A scheme for asynchronous data transfers which provides more efficient utilization of data-bus bandwidth and major simplification in the programming of data transfers has also been described. All data transfer latencies are automatically compensated by the built-in extrapolation algorithms.

The ADI SIMsystem provides the hardware, software, and distributed microprocessors to implement the asynchronous multi-processor methodology described in this paper. It is felt that this methodology is capable of achieving a major breakthrough in the design and operation of complex real-time simulations, particularly in the area of training systems.

References

1. Haraldsdottir, A., and R.M. Howe, "Multiple Frame-rate Integration," *Proc. of the AIAA Flight Simulation Technologies Conference*, Atlanta, Aug., 1988, pp 26-35.
2. Howe, R.M., "Dynamics of Digital Extrapolation and Interpolation," *Trans. of the Society for Computer Simulation*, Vol. 2, No. 1, Dec., 1985, pp 169-87.
3. Howe, R.M., *A System for Integrating Parallel Processors and Hardware in an Asynchronous Real-time Environment*, ADIUS Europe '92, Applied Dynamics International, 3800 Stone School Road, Ann Arbor, MI 48108, October, 1992.
4. Howe, R.M., "Some Methods for Reducing the Effect of Time Delays in Flight Simulation," *Proc. of the AIAA*

Flight Simulation Technologies Conference, Dayton, Ohio, Sept. 17-19, 1990, pp. 225-232.

5. Howe, R.M., "Real-time Simulation of Dynamic Systems Using a Variable Step Size," *Proceedings of the 1993 Simulation Multiconference*, Arlington, VA, March 29-April 1, 1993, pp 452-58.

6. Howe, R.M., *Real-time Simulation of Dynamic Systems Using a Variable Step Size*, ADIUS '93, Applied Dynamics International, 3800 Stone School Road, Ann Arbor, MI 48108, October, 1992.

7. Deiss, W.H., E.J. Fadden, and R.M. Howe, "Improving the Performance of Flight Simulators via Smart I/O Interface Systems," *Proc. of the AIAA Flight Simulation Technologies Conference*, New Orleans, Aug. 12-14, 1991, pp 93-115.

SEMI-FULL-SCALE DYNAMIC SIMULATION
COMPLEX ON THE BASIS OF CENTRIFUGE

V. M. Vasiletz
The Institute of Aviation and Space Medicine
Moscow, Russia

O. A. Yakimenko
Zhukovskiy Military Air Force Engineering Academy
Moscow, Russia

Abstract

The paper considers a structure and fundamental characteristics of semi-full-scale imitative dynamic simulation complex (IDSC) on the base of centrifuge. Capacities of the complex are illustrated graphically with some results of made on it different experimental works on investigation, optimization and ergonomic justification of performances of the closed loop "pilot-control system-aircraft-environment" with different composition of antistress facilities, different configuration of controls, action of different flight factors. It must be emphasized that represented results have rather particular character, because pursue an aim only to show IDSC capacities for imitative modeling of different investigation, design and test of flying machines (FM) specimens tasks, - so the details of definition and carrying of each particular experiment equally as its results in-depth discussion are not presented here, but may be a subject of further detailed consideration.

I. Fundamental complex characteristics

IDSC consists from an analog-digital computer complex (ADC) and centrifuge (CF).

The centrifuge has been equipped with a turned pilot station (a cockpit of centrifuge), which consists from a reclined seat configuration with capability to change the seat inclination within from 17 to 65 degrees, head up display, flying controls and a centrifuge emergency stopping lever.

The centrifuge has 7m arm and provides:

- a rotation angular rate maximum - 2π rps;
- a radial G-load maximum - 30;
- a tangential rate maximum - 180 km/h;
- a G-load holding accuracy - 0.1;
- a G-load onset rate maximum - 5 s^{-1} ;
- a G-load onset rate minimum - 0.1 s^{-1} .

The cockpit of centrifuge has two degrees of freedom.

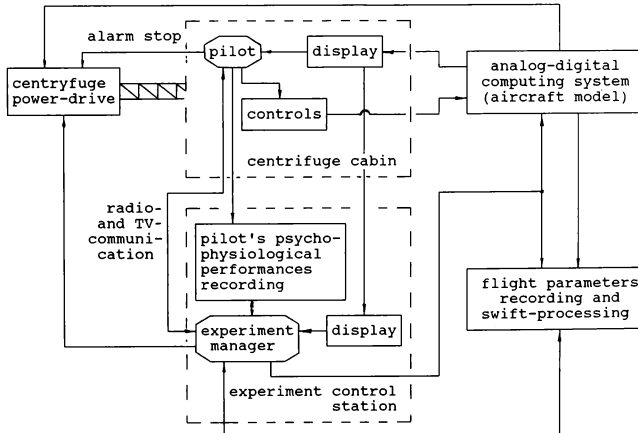


Fig. 1. Structural diagram of semi-scale dynamic complex.

The simulation complex includes a control panel for the experiment (training) team of medics and engineers, but also an equipment for recording and making express-evaluation of medico-physiological and technical data. The investigation (experiment) manager has a radio- and TV-communication with the cockpit operator. The scheme of the complex structure has been represented at Fig.1.

The CF-operating conditions (stress profile) may be given by a program generator and also may be determined by a pilot-operator on the basis of a control deflection (aeroplane control stick on longitudinal direction).

ADC software allows to simulate in the real time (with discreteness 50 s) flight vehicle motion, which the CF-cockpit operator guides by appropriate controls displacements while determining of aeroplane attitude the apparatus of quaternions is used. Data bank contains characteristics of aeroplane aerodynamic, engine power, system of stability and controllability improvement, mass-inertial and grouping parameters, needed for simulation dependences of aerodynamic coefficients from a flight regime and FM-configuration.

The analog part of the complex allows to give different loaded diagrams of the controls (control system structure and parameters).

Centrifuge amplitude-phase-frequency response (APFR) at the different stress base values and an APFR of aeroplane model (for fixed flight regime) are shown on Fig.2.

ADC permits to modelling both pilotage and air target attack tasks (the movement of target have many variations and is given by a program generator).

The comparison of normalized spectral stress signal densities in modelling on centrifuge and real typical training air battles (look at Fig.3, where spectral densities of aeroplane, which attacks manoeuvring with different steady-state stresses $n_{y,ef}$ air target, are given) with centrifuge APFR (Fig.2) gives a fundament to regard the results of modelling as reliable and a centrifuge - in work frequency range - as linear object.

Antistress pilot's facilities consists of an oxygen-breathing apparatus, which supply a breath under excess pressure, an antistress suit, controlled by pressure automaton, and a protective hard hat with an oxygen mask.

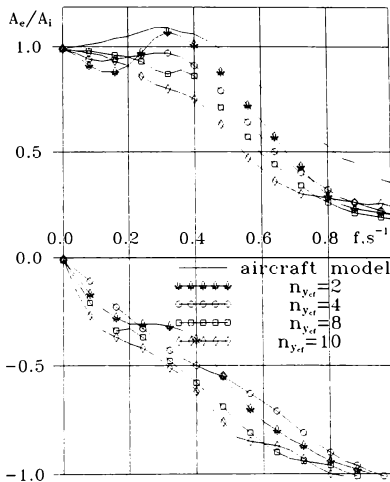


Fig. 2. APFR of centrifuge ($n_{y,ef}=2,4,8,10$) and aircraft model.

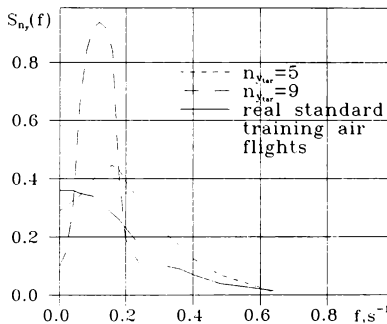


Fig. 3. Normalized spectral density of stress signal.

Before, during and after experiments such indices of operator's psychophysiological state as heart rate, respiration rate, stroke volume (it is evaluated by tetrapolar rheocardiography method), a systolic pressure in earlap vessels with a photoplethysmography have been recorded and analysed.

During experiments an electromyography activity of muscles, a displacement rate of head, trunk and pelvis (during the exposure of lateral and combined G-loads) also could be determined.

It is also realized a recording the indices of flying performance, such as holding accuracy of preset flight parameters (a tracking by director marker glide-path indicator), frequency characteristics of pilot handling actions on the controls et al.

The IDSC is used as a simulator for selection, education and training of aircrews and cosmonauts, but also allows to perform experimental investigations on next directions:

- search of optimal characteristics of an aircraft stability and controllability, providing more effective task performance;
- examination of flight control effectiveness, depending on an orientation of G-load total vector concerning coupled system coordinate axes;
- evaluation of pilot's psychophysiological state on the efficiency of mission execution (aerobatic flying, attitude flying, keeping of specified parameters for a landing or an in-flight refuelling);
- examination of efficiency of some methods and means, which improve a pilot G-tolerance limit (including an antistress equipment, the reclined seat et al.);
- examination of a composition and characteristics of instrument equipment, that is needed for flying activity (including a multi-mode display);
- investigation of operational effectiveness of a perspective flight controls (including the strain-gauge pedal, an aircraft side control stick, an engine control stick);
- examination of effects of G-loads and the controls pattern on thresholds of pilot control actions;
- evaluation of being developed control systems, adaptive to a flying regime and pilot characteristics (including joined control system) and others.

Obtaining experiments data provides to form ergonomic-justified requirements to the image of FM, its systems structure and parameters and also given abounding materials for mathematic modelling.

II. Particular results of some experimental investigations

The most well-understandable way to demonstrate IDSC facilities is to present a number of specific experimental investigations results.

Evaluation of aeroplane dynamic perfection

ADC permits to investigate own aeroplane characteristics in whole range of flight altitudes and velocities (on Fig.4, as example, are given a lines of equal meanings of α -coefficient, which represents a degree of roll and yaw control channels interaction). If further inverse altitude-velocity range and display it in the plane of two characteristic movement parameters (at Fig.5 as such parameters are chosen a base frequency and relative damping coefficient of "small" longitudinal movement oscillations), and also apply on it familiar (previously received by help of IDSC) regions of pilot's pilotage estimations (on Cuper-Harper scale, for example), there will be well-understandable then whether (and where) the dynamic characteristics of being investigating aeroplane satisfy to imposed requirements for it.

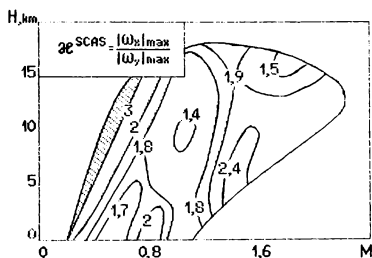


Fig. 4. 2α equal meanings curves.

Definition of aeroplane controllability efficiency under the combined stresses acting

Combined stresses IDSC-modelling methods (by the aid of cabin fixation) is shown on Fig.6. And Fig.7,8 demonstrate a tolerable times of different combined stresses action on a pilot, received at process of extensive experiment, and equipotentials of the relative probability of successful flight task execution by a pilot in dependence of value and direction of acting stress.

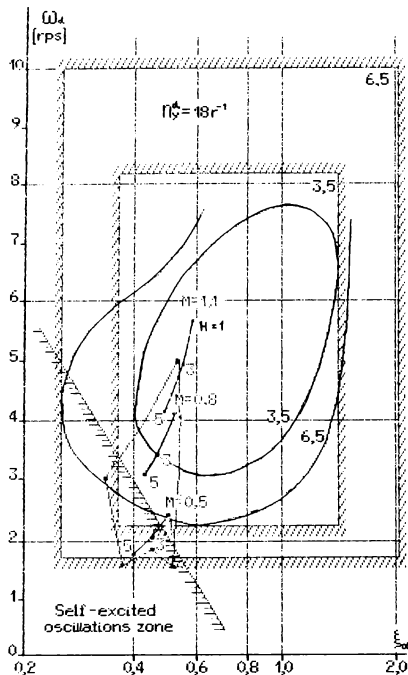


Fig. 5. Pilot's handling estimations integrated with altitude-velocity flight range in the plane of the small longitudinal movement parameters ($n_y = \text{const}$).

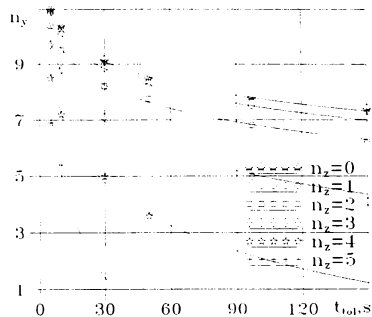


Fig. 7. Tolerable time of different combined stresses action on a pilot.

Justification of perspective controls application efficiency

Side stick advantages over the central stick in the tasks, which require an accurate aeroplane handling, are today already well-known.

IDSC permitted, among others, to obtain an ergonomic justification of dimension, location, mobility and loaded diagrams of side stick. Fig.9 demonstrates balance of side stick control movements in longitudinal and lateral channels.

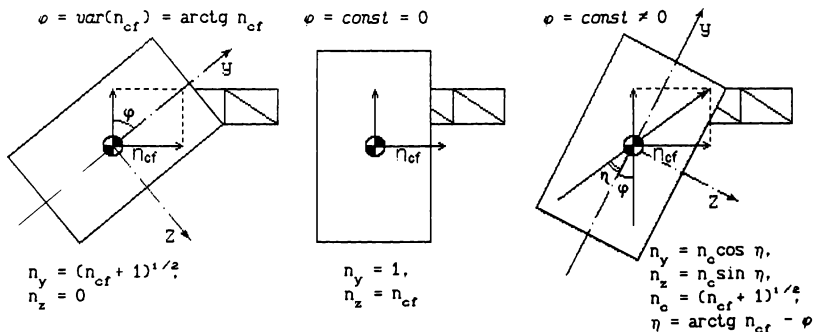


Fig. 6. Combined stresses modelling.

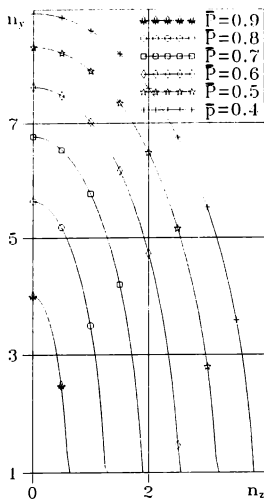


Fig. 8. Relative probability of successful task execution by a pilot in dependence from value and direction of acting stress.

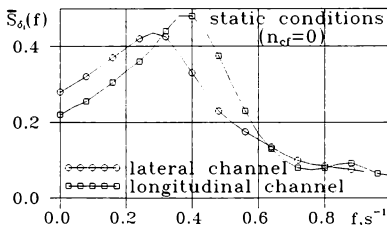


Fig. 9. Normalized spectral density of control movements.

Adaptive control system forming

In the aim of forming the structure and definition parameters of control system, adaptive to changing according to flight conditions pilot's psychophysiological state control system, by help of IDSC were investigated the dependences of threshold pilot's forces at controls from flight conditions.

The further data processing by the aid of principles of inferential statistics allowed to obtain the fundamental results similar to ones on Fig.10,11. Appropriate interpretation of them permitted to design an adaptive control system (on Fig.12 is schematically shown only part of cross channel of such system), which demonstrated on IDSC 25% growth of hand-operating quality under the high G-stress action.

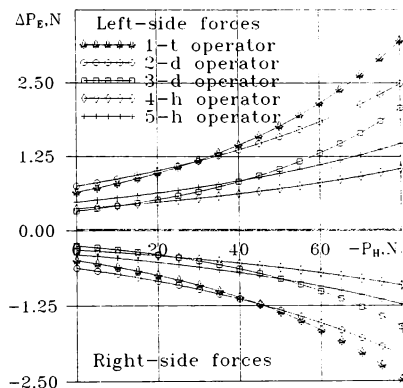


Fig. 10. Dependences of absolute threshold pilot's forces on the side stick in the lateral channel from balanced force on it in the longitudinal channel.

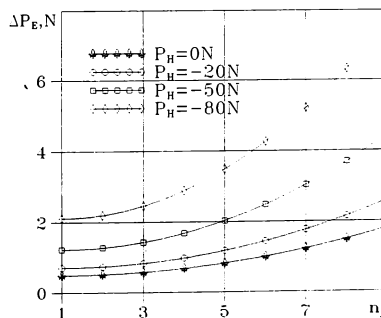


Fig. 11. Generalized dependence of the same (as on Fig.10) parameter from acting normal stress.

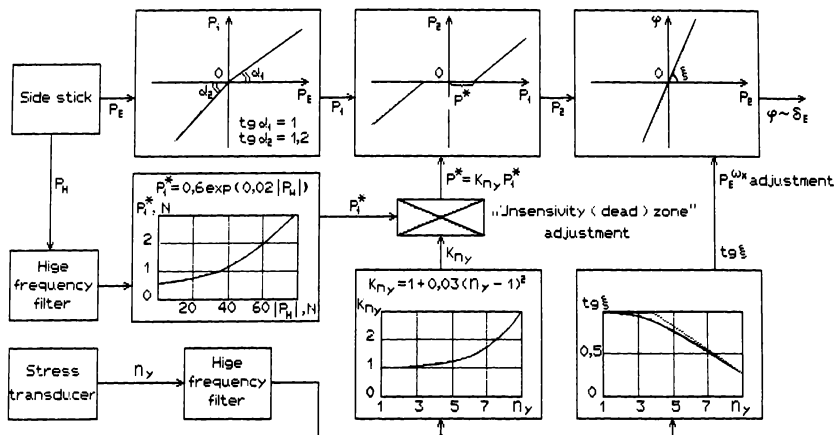


Fig. 12. Structural scheme of the cross channel of adaptive to flight conditions control system.

Conclusion

Thus, considered semi-full-scale imitative dynamic simulation complex on the base of centrifuge permits to carry out fairly extensive circuit of investigations with

pilot in control contour, what graphically visualized by represented in this paper results of a number of experimental works, which reliability was verified in rather more expensive flight experiments more than once.

FALSE CUE DETECTION THRESHOLDS IN FLIGHT SIMULATION

by

Ruud J.A.W. Hosman and Han F.A.M. van der Steen
 Faculty of Aerospace Engineering
 Delft University of Technology
 Delft, The Netherlands

Abstract

In this paper a new experimental method is presented for the design of motion filters, based on simple concepts using a multi sensory perception model.

We will present the results of recent research on the perception thresholds for differences in the visual and vestibular cues as determined by using a sled as well as a research flight simulator for linear motions (surge and heave). These differences will be called *false cues*. Peripheral visual cues are modulated relative to the vestibular cues to determine the threshold values. The thresholds turn out to be dependent on velocity magnitude. The no-motion ranges turn out to be so large that they are useful in simulation.

Introduction

Simulation of aircraft motions has been a topic in flight simulation since in the early seventies when six-degrees of freedom motion systems were widely introduced. Because simulator motions cannot be equal to the actual motions of the simulated aircraft, motion filters have to transform the aircraft motions to maintain the simulator motion system within its mechanical limitations. The rationale behind these motion filters is that the errors or false cues have to be small enough not to be recognized by the pilot. In that sense, linear and rotational acceleration errors should be smaller than the thresholds of vestibular motion perception. However, this cannot be achieved under all circumstances due to the very low values of the vestibular thresholds. Research in Delft has shown that under high mental load conditions (a normal situation in flight simulation training), vestibular thresholds are increased considerably¹. These higher thresholds allow for the passing of larger motion errors which turns out to be useful. The technique usually applied is to optimize the parameters of the motion filters in such a way that the vestibular stimulation of the pilot in the simulator is (as much as possible) equal to the vestibular stimulation in the aircraft^{2,3}. Until now the development of motion filters has been based on the aforementioned concept and quite reasonable results have been obtained in most cases. The theoretical and experimental basis for the design of motion filters, however, was developed in the seventies and until now, no real further developments of this basis have occurred. Due to the rapid improvement of simulation technology over the last twenty years, the application of the flight

simulator made a tremendous impact as a training device. Loft training, proficiency checks and full type conversion training became the standard in flight simulation training. This development has urged the authorities to set up requirements for various fidelity levels of flight simulation⁴. In these requirements certain measures are given with respect to the motion simulation. Although for certain standards, as for instance Level A, B, C and D, the capabilities of the motion system are specified, the requirements do not specify the fidelity of the generated motion as such.

An important reason for the lack of attention and requirements on the motion generation is the fast development of computer-generated 'outside world displays'. From the simple black and white night scene displays with limited field of view these displays evolved to color daylight wide angle displays which provide the pilot with strong visual motion cues. Based on research at the Faculty of Aerospace Engineering of the Delft University of Technology (DUT) on visual vestibular perception of motion⁵, it is clear that in the presence of strong visual cues, vestibular motion cues still provide important information to the pilot. In our view, the perception of aircraft motions by the pilot is a complicated process. As will be shown in this paper in the perception process, visual and vestibular cues interact at different levels.

A few years ago a number of research institutes in the Netherlands, among others TNO Institute for Perception and the Delft University of Technology, began a research program on the Space Adaptation Syndrome (SAS) or space motion sickness. In this program an experimental technique was developed to establish changes in the vestibular motion perception due to an adaptation to a +3G environment in a centrifuge⁶. This technique uses the visual motion perception as a reference system relative to the vestibular motion perception and is based on a functional model for motion perception⁷. It turned out that this technique could be used to determine the thresholds for perception of differences between visual and vestibular-perceived motion. The results obtained showed that between these thresholds a rather large "no-motion" range exists wherein subjects cannot detect differences between visual and vestibular motion stimuli. This is an important result because in flight simulation there will always appear mismatches between the visual and vestibular motion cues. If the no-motion ranges could be determined for the linear and rotational motion perception cues, then the motion filter optimization could be reviewed.

The technique was first evaluated with a sled of the European Space Agency (ESA) which was made available by ESA to the above mentioned group of research institutes in the Netherlands. After the feasibility of the technique for surge motion was established, an experiment on heave and roll motion was performed with the Delft research simulator with its three degrees of freedom motion system. Given the results obtained with these experiments, the National Aerospace Laboratory (NLR) in Amsterdam supported this research by making their six-degrees-of-freedom motion system available for experiments with surge, heave, roll and yaw motions.

The present paper deals with the experimental research performed with linear motion using the ESA sled, the Delft flight simulator and the NLR six-degrees-of-freedom motion system. In the next paragraph the theoretical background for this research is presented. After that, the experimental procedure and the experiments will be explained. The paper will be concluded with a discussion of these results.

Theory

The perception of motion results from a number of different signals which converge from several sensory systems to one percept of motion in the brain. Although the same signals take part in the perception of object motion and self- or egomotion, a distinction has to be made between these two perceptions. This can be demonstrated by the functional model for motion perception, see figure 1, as proposed by Wertheim⁷. This model shows both the relationship between the perception of object motion and the perception of egomotion, and the generation of reference and retinal signals. Based on this model and the definitions of these signals, the thresholds for a disagreement between the visually and vestibularly perceived motion will be defined.

The inputs to this model are the stimulus of moving images to the retina, the head movements and the observers decision or "will" to move the eyes relative to the head. The visual system transforms the movement of the image on the retina to an estimate of the retinal image velocity V_{ret} . This velocity has to be compared with an other velocity signal, the velocity of the retina surface, i.e. the eye velocity in the fixed world. This velocity signal will be referred to as the reference velocity V_{ref} . This reference velocity is the result of the eye movements relative to the head and the head movement relative to the fixed world.

The eye movement as controlled by the oculomotor system results from the "will" to direct our eyes in a certain direction. The *effference copy* is the output of the eye velocity estimator and is a copy of the feedback signal of the oculomotor system.

The head movement in space is determined by an estimator using the vestibular output and the optokinetic visual output. The vestibular system is sensitive for linear and angular acceleration. Besides the otoliths and

the semicircular canals which form the vestibular system, somatosensory (pressure on the skin) and proprioception (pressures in the internal body) information can also be detected. We will however combine all of these information sources and call it the vestibular system.

The estimated velocity based on the vestibular stimulus is accurate over a rather short time span and is corrected by the visual optokinetic stimulus. Thus, under normal conditions, the estimated head velocity in space depends on both the vestibular output and the optokinetic output.

When we move around in our stationary world the retinal image velocity V_{ret} is opposite and equal to the reference velocity V_{ref} . So:

$$V_{ret} = V_{ref}$$

In the real life situation however, both velocity signals will be corrupted with perception errors and noise. Thus the above relation has to be modified to:

$$V_{ret} \approx V_{ref}$$

In flight simulation, the visual and vestibular cues should be chosen in such a way that this relation remains valid. Then, false cue perception will not occur.

From previous research^{7,8} it is clear that a difference in magnitude between the retinal velocity and the reference velocity can only be detected if this difference exceeds a certain minimum value. This minimum value is called the Just Noticeable Difference (JND) of retinal and reference velocity. As long as:

$$|V_{ret} - V_{ref}| < \text{JND}$$

both velocities will be considered equal and the environment is perceived stationary.

Now imagine the case of a simulator where the low linear velocity in a certain direction of the cab differs from the linear velocity as displayed on the visual system along the same direction. As soon as the difference between the reference velocities equals the JND, the threshold is reached and the pilot will perceive a conflict. When the visual surround is moving with the simulator, the magnitude of the retinal image threshold velocity is

$$V_{ret,with} = V_{ref} - \text{JND}$$

and when the visual surround is moving against the simulator cab the magnitude is

$$V_{ret,against} = V_{ref} + \text{JND}$$

When the two thresholds have been determined, the actual value of the JND follows with:

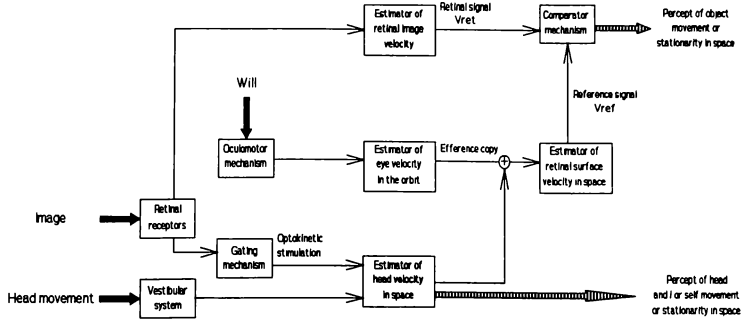


Figure 1: Functional model describing the perception of object and self-motion adapted from Wertheim⁷.

$$JND = \frac{1}{2} (V_{ret,against} - V_{ret,with})$$

The point just in between the two thresholds is considered as the Point of Subjective Stationarity (PSS). At this point the retinal image velocity $V_{ret,PSS}$ corresponds with the reference velocity V_{ref} , thus:

$$V_{ref} = \frac{1}{2} (V_{ret,against} + V_{ret,with}) = V_{ret,PSS}$$

Finally the gain G of the reference velocity can be defined. This gain is the ratio between the reference velocity V_{ref} as estimated by the perception process and the actual velocity of the retinal surface V_{eyes} in space:

$$G = V_{ref} / V_{eyes}$$

Where V_{ref} is determined relative to V_{ret} , the definition of the gain G is correct only if it is assumed that the gain of the retinal velocity estimator is equal to one. This is a reasonable assumption because the perception of long-term movements in every day life is correct, thus the gain should be one.

In figure 2, the relation between the threshold values, the gain, the PSS and the no-motion range are presented. If the head and eyes are fixed relative to the motion system, then the velocity of the eyes is equal to that of the motion system:

$$V_{eyes} = V_{mot}$$

In simulation, the flight instruments and the outside world displays provide the pilot with visual stimuli about the aircraft motion in a simulated stationary world.

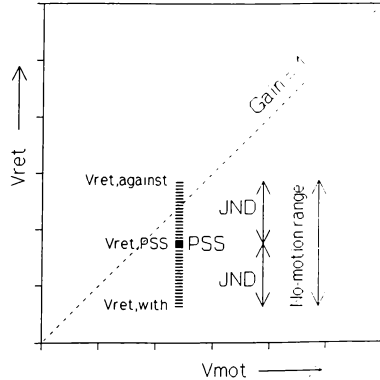


Figure 2: The concept of JND, PSS, gain and no-motion range.

Although these stimuli are not fixed to the real stationary world, the pilot is expected to accept that these stimuli represent a real stationary world. Thus retinal reference velocity as perceived in the simulator, $V_{ret,sim}$, is considered as a reliable reference signal. The reference velocity signal $V_{ref,sim}$ as far as it is based on the vestibular input due to simulator motion, may differ consider-

ably from the simulated aircraft velocity and consequently from $V_{ref, sim}$. If the difference exceeds the JND or if $V_{ref, sim}$ is outside the no-motion range a conflict is perceived by the simulator pilot. This conflict will normally be attributed to the motion cues. Thus the magnitude of the no-motion range is directly related to the perception capability of "false motion cues".

The next question which arises is how to determine the JND and the gain G of the reference velocity. The functional perception model as presented in figure 1 shows three different inputs to the model; retinal image velocity, eye velocity as commanded by our "will" and head movement relative to the stationary world or *space*, how it is called in the model. When eye movement relative to the head and head movement relative to the motion system is excluded and optokinetic stimulation is prevented, then the resulting reference velocity V_{ref} is based only on the vestibular system input due to motion of the motion system. In that case the model in figure 1 can be reduced to a much simpler model as presented in figure 3. The only inputs are the movement of patterns in the visual field stimulating the retina and the movement of the motion system stimulating the vestibular system. The output of the model under these simplified conditions is the movement of the patterns relative to the stationary world.

Now imagine an experiment in which the subject is moved sinusoidally by a motion system. At a fixed position, for instance at maximum velocity, a moving pattern is presented to the subject. The subject is asked to decide whether this pattern is moving or stationary relative to the stationary world. If the pattern is perceived to move relative to the stationary world, the subject perceives object or pattern motion. If the pattern is perceived as being stationary, it represents the fixed world. Given the above defined no-motion range it is the aim of the experiment to find the with- and against-thresholds by changing the pattern velocity every period of the sinusoid. As shown before, the gain and the no-motion range can be determined from these thresholds. The experimental method proposed here has the advantage that a relative simple experimental setup can be used. The method in itself, however, limits the range of the frequency and amplitude which can be used. The frequency is limited by the response time of the subjects and the amplitude is limited by the maximum stroke of the motion system used.

In the next section, three experiments are described to determine the gain and the JND for linear surge and heave motion. The main reason for starting with linear motion was that the first evaluation experiment was performed with the ESA sled which only can move along a linear track. It does however has the advantage that the otoliths, the organs of the vestibular apparatus which are sensitive to specific forces, have an approximately constant modulus and phase angle in the frequency range that is applied. Therefore it may be assumed that

the dynamic characteristics of the otoliths do not have any influence on the results.

Method

An experimental procedure has to be designed to test the match of the reference and the retinal signal as explained in the theory. The reference signal will be induced by an applied motion to a chair in which the subject is seated. This motion must be employed in full darkness to exclude any visual cues which could influence the reference signal in an optokinetic or cognitive way. The retinal signal is generated by monitors. These will stimulate the peripheral part of the visual field by checkerboard patterns that move on the displays, along the same direction as the direction of the applied motion. The method runs as follows. The subject moves along one direction with a sinusoidal motion profile. The maximum acceleration of the applied motion is chosen to be well above the vestibular threshold so that a reference signal is generated. When the subject reaches the maximum velocity of the oscillation, around the top of the sine, the monitors briefly display the patterns in the peripheral visual field at eye level. The maximum velocity occurs twice during such an oscillation; maximum positive and maximum negative velocity. Both cases will be tested in separate sessions.

The task of the subject is to judge whether these patterns move or are standing still relative to the outside (fixed) world. If such a relative motion is perceived, the subject has to answer in what direction this motion occurs. In this way we can investigate how the subject perceives the applied body motion in terms of a corresponding peripheral optic flow; the match of reference and retinal signal.

The motion parameter we will test is the *velocity* of the motion. Studies showed that although the vestibular organs are only sensitive to specific forces and rotational accelerations, neural processing leads to a velocity dependent afferent signal in the natural frequency and acceleration range of oscillations¹. Since the size of the retinal signal is also velocity dependent⁷, we have two signals of equal dimension and can compare these when we do experiments in this range.

To analyze the velocity dependency of the perception, we must vary the signals involved. The most evident way to do so is by keeping one signal fixed and varying the other. In our experiments, we keep the reference signal fixed and vary the retinal signal. The reason for the variation of the retinal signal instead of the reference signal stems from the consideration that we have to keep in mind that the reference signal is generated by mechanoreceptors and is therefore subject to certain dynamic system properties, like transients and threshold values. The influence of such system parameters can only be dealt with if an appropriate motion profile is applied. By using an elementary sine profile that is of a constant form throughout a session, these parameters

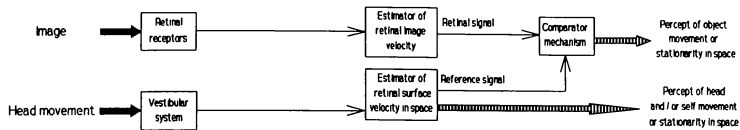


Figure 3: The same model as in figure 1 but with some simplifications: no eye movements relative to the head are made and the optokinetic component is excluded. This will be the model on which our experiments are based.

are easily described and accounted for. A reason of a more practical origin is the easy way in which the visual stimulus can be altered with respect to the motion stimulus. In the sled experiments, the sled motion profile cannot be changed during a session, so we must vary the retinal signal. Besides that, we will then be able to compare the results of the sled with those of the moving base experiments.

The display time of the checkerboard patterns is chosen to be 0.40 s. Short enough to exclude optokinetic stimulation¹⁰ and long enough to extract the displayed retinal velocity accurately.⁵

Because linear motions are made, the visual flow is distance dependent. E.g. when moving linear with a certain velocity, a nearby part of the visual field will move faster than a part that is far away from the observer. So we have to instruct the subject well about the distance of the patterns that will be seen. The subject has to know this distance to make an appropriate deduction of what the velocity on the monitors that will be displayed represents. The match of the reference signal to the retinal signal is therefore completely dependent of this distance gauging. A careful instruction will be made to get this estimation right.

We now have designed an experimental method by which a whole range of velocities can be tested and the parameters that are described in the theory (gain and no-motion range) can be found.

Experiments

Two kinds of motion systems are used to study the accordance of visual and non-visual motion input; a sled on a linear track and two research moving bases.

In the experimental set up of the ESA sled, the reference signal is induced by the motion of this sled, moving on a linear track. The subject is seated with the head fixed to the chair. The motion is perpendicular to the frontal plane of the subject (surge motion) and takes place in darkness. During the motion, the subject is instructed to fixate on a small light source (an LED) at the end of the track. Through this fixation, no eye movements are made apart from very small vergence move-

ments. The retinal signal is induced by patterns on monitors, placed alongside the track at eye level and attached to the laboratory floor.

The other two experiments were conducted with moving bases: the DUT flight simulator and the NLR moving base. In these experiments, the subject is inside a cockpit in a normal pilot's seat. The head of the subject is, unlike the sled experiments, *not* fixed to the chair. The motion profile is chosen to be perpendicular to the frontal (surge) or to the transversal plane (heave) of the subject. The retinal signal is induced by monitors at eye level which are fixed on the simulator in the periphery of the subject's visual field. The position of the monitors is chosen to be equal to the position in the sled experiments at the stimulus presentation time. The subject is instructed to fixate on a small light source fixed in the cockpit which is in front of the subject. Through this fixation, no eye movements are made during the motion. An important difference with the sled experiments lies in the interpretation of the visual stimulus. In contrast with the sled experiment, now the monitors are moving *with* the subject. The instruction to the subject is that the patterns which will be briefly seen represent the fixed outside world. The fact that the subject has to keep in mind that the outside world will visually be presented and has to be judged, causes a careful instruction to the subject. The task of the subjects is to judge whether they see motion of this outside world or not and in which direction. The extra instruction to the subject is made that the distance of the patterns which will be presented are *directly* behind the virtual windows of the cabin, and the distance of the stimulus then equals the distance of the monitors in the sled experiments.

We can scan the range of the perception of differences between retinal and reference signals by the subject in several ways. We have chosen to use the PEST-algorithm⁹ for an active search for the borders of the no-motion ranges; the thresholds of the velocity area in which the reference and the retinal signal are judged to be in concordance. The application of this algorithm is done by a 3-alternative forced choice task. E.g. in the experiment where the subject moves back and forth, the alternatives are: the checkerboard patterns are moving *forward* relative to the fixed world, the patterns move

backward or the patterns are standing *still*. If the subject can't make a decision, the "still" interpretation is to be chosen. The subject has three buttons to make these answers which can be pressed and will be hold in the hands during the session. The subject has to make a decision in a quarter of a period of the oscillation time, starting at the time where the visual stimulus, around maximum velocity, is presented. If the answer is too late, the algorithm reports this and the next visual stimulus will be equal to the presented one.

Motion profiles

The profiles are chosen to be in a natural range of body and head movements. We therefore have chosen circular frequencies from 0.71 to 2.0 rads^{-1} and amplitudes to reach the natural range so that we remain in the velocity range of the vestibular system. The velocities of the motion vary from 0.26 to 0.50 ms^{-1} in the heave experiments, and from 0.26 to 1.10 ms^{-1} in the surge experiments.

In order to avoid the influence of transients in the experiment, the oscillation is applied for several periods after which the measurement starts. In the experiments using the moving bases, the amplitude of the motion is also gradually built up to its intended value (over two periods) to avoid large accelerations of the simulator at the start of the motion. The algorithm that computes the sled motion does not build up to an amplitude but avoids large accelerations by a smooth start of the motion.

Subjects

Large groups of subjects (12 to 18 students) aged 18 - 26 years were selected for the experiments. The same subjects were used in the experimental setups at a location.

Results

In the figures 4 to 10 the results of the experiments are presented. The different results are denoted by an *S* for the ESA sled, a *D* for the DUT simulator and an *A* for the NLR simulator. The PSS is denoted with a solid square, the dashed line represents the no-motion range. From figures 4 and 5 can be seen that the NLR experiment fits well between the ESA experiments. The gain decreases with increasing velocity. This is also the case in the heave experiments: figure 6 and 7. Both for the heave and the surge experiments the value $G = 1$ is covered by the no-motion range.

From the plot of the no-motion ranges as a function of V_{mot} , figures 8 and 9 it is clear that this range increases with increasing motion (reference) signal. The variance is large for the NLR experiments. When we take a look at the individual results of the subjects for one of the sessions (figure 10, where $V_{\text{mot}} = 1.10 \text{ ms}^{-1}$, experiment

with the ESA sled) we see apparent individual variations. In this figure, a sample of the subjects is drawn.

Discussion

The results of the surge experiments indicate that the experimental method does not depend on the way in which the visual stimulus is presented: the gain of the NLR experiments fits between those obtained from the ESA sled (figures 4 and 5). This implies that results of sled experiments can be regarded as useful material for motion filter research.

The heave experiments do not differ much from the surge experiments (figures 4 to 7). The decreasing gain at higher velocities of the motion system are alike. It is remarkable that the gain decreases with increasing size of the applied velocity. This means that at higher velocities, the signal from the motion stimulus is underestimated related to the visual stimulus. But although there is a decreasing gain, the mean no-motion range always overlaps the case of a gain equal to one. A correct interpretation of the relation between the motion and visual stimuli is still present.

A further correspondence is the increasing no-motion ranges in surge and heave. The size of the no-motion range increases with increasing signal size but not linearly which we would expect according to Weber's law which says that the signal noise is proportional to the size of the signal. If statistics is skipped from consideration, this could imply that velocity is not the only perception signal involved. Further analysis will have to be done to elicit such a component.

The no-motion ranges at particular velocities are larger when the subject moves backward related to the forward condition in the surge experiments (figure 4 and 5). This implies that the signal noise is larger for this motion type, which can be due to the less natural kind of motion. The human perception system is apparently less accurately equipped for the backward type of motion, due to the lack of experience with the backward motion.

Since the DUT experiments show no such large deviation as the NLR experiments (figures 8 and 9), the large deviation in the NLR experiment probably does not arise from the different way in which the visual stimulus is presented; monitors moving with the subject instead of fixed monitors along the track of the sled. It is more likely that this is due to the specific group of subjects.

The individual results which are drawn in figure 10 indicate that the no-motion ranges and the gain of the signal are obvious individual properties. The sensitivity to noise of the experimental method, for instance the errors in distance gauging of the patterns by the subject, can however contribute to these individual differences.

The idea of a gain of the retinal signal equal to one what we assumed in our theory is perhaps only appropriate if the whole visual field is stimulated instead of the peripheral part what we did. However we think that this kind of stimulation does not vary much that we have to

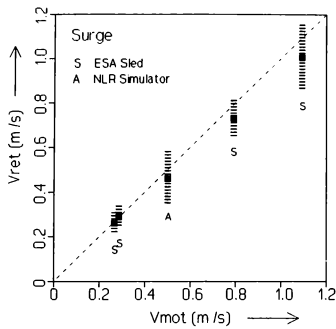


Figure 4: Surge experiments. The visual stimulus is presented when the subject moves forward.

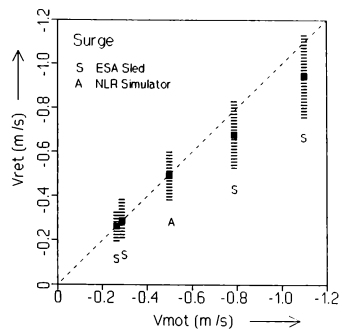


Figure 5: Surge experiments. The visual stimulus is presented when the subject moves backward.

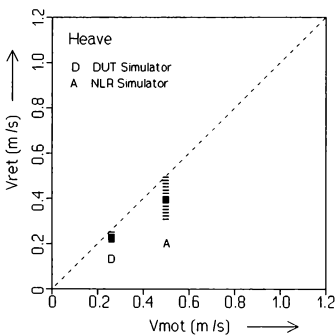


Figure 6: Heave experiments. The visual stimulus is presented when the subject moves upward.

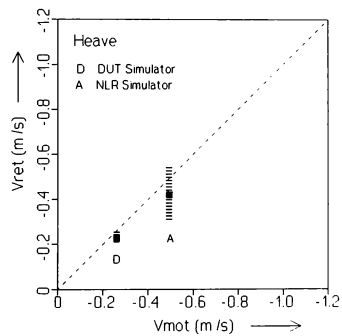


Figure 7: Heave experiments. The visual stimulus is presented when the subject moves downward.

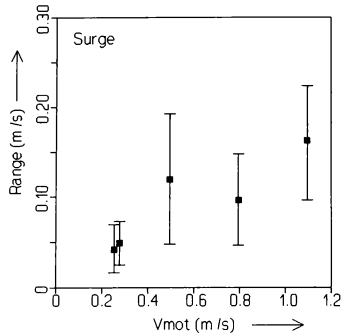


Figure 8: The no-motion ranges and their variances in the surge experiments. The visual stimulus is presented when the subject moves forward.

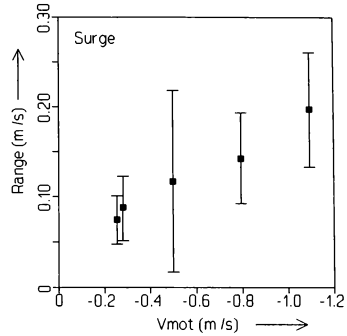


Figure 9: No-motion ranges and their variances in the surge experiments. The visual stimulus is presented when the subject move backward.

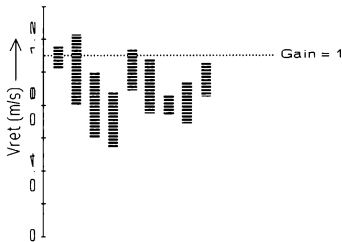


Figure 10: Surge experiments with the ESA sled. A sample of the results of individual subjects is drawn. The visual stimulus is presented when the subject moves forward.

reconsider the assumption of that gain, it is better when the whole visual field is stimulated. The outside world displays can offer us such a possibility.

Conclusions

The most important result we found is that the experimental method to study the thresholds between visual and vestibular correspondent signals can be applied on a moving base system. This means that this method can be developed further and a new evaluation of the implementation of certain thresholds in motion filters can be reviewed.

We have shown that there is no such one-to-one process when dealing with human motion perception using the described inputs. Just using the knowledge about vestibular perception is a rather poor start to describe the perception of motion by the pilot. The question arises in what manner further research on optimizing motion filters can be done. Our experiments open a new insight and we think that it will contribute to a better performance on the long term.

Although a lot of work remains to be done, we have described a basis on which further research can proceed.

Acknowledgments

ESA is acknowledged for providing one of their sleds for this research.

The National Aerospace Laboratory *NLR* in Amsterdam for providing access to their 6-DOF simulator motion system.

The Netherlands Organization for Scientific Research *NWO* for partly sponsoring this research.

References

1. Hosman RJAW and Vaart JC van der (1978). *Vestibular Models and Thresholds of Motion Perception. Results of Tests in a Flight Simulator*. Delft University of Technology, Faculty of Aerospace Engineering. Report LR-265, Delft.
2. Sinacori JB (1973). *A Practical Approach to Motion Stimulation*. AIAA Paper 73-931.
3. Bussolari SR, Young LR and Lee AT (1989). *The Use of Vestibular Models for Decision and Evaluation of Flight Simulator Motion*. AIAA Paper 89-3274-CP.
4. Anon (1989). *Airplane Simulator Qualification*. Federal Aviation Administration Advisory Circular AC No. 120-40B.
5. Hosman RJAW and Vaart JC van der (1989). *Visual-Vestibular Interaction in Pilot's Perception of Aircraft or Simulator Motion*. AIAA Paper 88-4622-CP.
6. Bles W, Bos JE, Furrer R, De Graaf B, Hosman RJAW, Kortschot HW, Krol JR, Kuipers A, Marcus JT, Messerschmidt E, Ockels WJ, Oosterveld WJ, Smit J, Wertheim AH and Wientjes CJE (1989). *Space Adapta-*

tion Syndrome induced by a long duration +3Gx centrifuge run. TNO Institute for Perception. Report IZF 1989-25, Soesterberg.

7. Wertheim AH (1990). *Visual, vestibular and oculomotor interactions in the perception of object motion during egomotion*. In Warren R and Wertheim AH (Eds), *Perception and Control of Self-Motion* (pp. 171-216). Lawrence Erlbaum, Hillsdale, NJ.

8. Wertheim AH (1981). *On the relativity of perceived motion*. *Acta Psychologica*, vol. 48, pp 97-110.

9. Taylor MM and Creelman CD (1967). *PEST: Efficient estimates on probability functions*. *Journal of the Acoustical Society of America*, vol 41, nr 4.

10. Howard IP (1982). *Human Visual Orientation*. New York: Wiley.

SIMULATION MOTION EFFECTS ON SINGLE AXIS COMPENSATORY TRACKING

Jeffery A. Schroeder*
NASA Ames Research Center
Moffett Field, California

Abstract

An experiment that examined how changes in a motion platform drive filter affect pilot-vehicle performance and opinion was conducted on the NASA Ames Vertical Motion Simulator. Pilots controlled a simplified helicopter model in the vertical or the directional axis and tracked a randomly moving target aircraft in the presence of a random disturbance. With both tasks, variations from full motion to fixed-base conditions were made in the high-pass drive filter gain and natural frequency. The results indicate that vertical motion did not affect the open-loop pilot-vehicle target-tracking crossover frequency, but target-tracking phase margins improved with increased filter gain or decreased natural frequency. Vertical disturbance-rejection crossover frequency increased with decreasing filter natural frequency, while disturbance rejection phase margins improved with increasing filter gain. Vertical tracking errors increased significantly when all vertical motion was removed. No significant differences were measured among the directional configurations, which indicates that pure yaw motion cues may not be as important as previously thought in flight simulation.

Notation

e	displayed target error, ft or deg
g	gravitational acceleration, ft/sec ²
h	vehicle model altitude, ft
\ddot{h}	vehicle model vertical acceleration, ft/sec ²
\ddot{h}_{filt}	motion system vertical acceleration command, ft/sec ²
\ddot{h}_{sim}	motion system vertical acceleration, ft/sec ²
h_{v}	vertical target position, ft
h_r	directional target position, deg
j	$\sqrt{-1}$
K	motion filter gain, nondimensional
s	complex variable, 1/sec

δ_c	cockpit collective position, in
δ_{c_d}	vertical disturbance input, in
$\delta_{c_{\text{tot}}}$	collective position plus disturbance, in
δ_r	cockpit pedal position, in
δ_{r_d}	directional disturbance input, in
$\delta_{r_{\text{tot}}}$	pedal position plus disturbance, in
ζ	motion filter damping ratio, nondimensional
σ	standard deviation
Φ_m	phase margin, deg
ψ	vehicle model yaw angle, deg
$\ddot{\psi}$	vehicle model yaw acceleration, deg/sec ²
$\ddot{\psi}_{\text{filt}}$	motion system yaw acceleration command, deg/sec ²
$\ddot{\psi}_{\text{sim}}$	motion system yaw acceleration, deg/sec ²
ω	frequency, rad/sec
ω_n	motion filter natural frequency, rad/sec
ω_c	crossover frequency, rad/sec

Introduction

To limit the displacements of ground-based motion simulators, the attenuation of math model accelerations is required for most maneuvers. Often, the attenuation is achieved by an ad hoc adjustment of the gain and natural frequency of a high-pass filter, which is applied to the math model accelerations. The resulting effects of these adjustments on a simulation's purpose, such as systems development or crew training, are not fully understood.¹⁻⁴ The lack of a definitive answer to these motion attenuation effects is not a result of a scarcity of effort, but it is due to the many possible variables that may influence the answer. These variables include the vehicle dynamics, task purpose and dynamics, visual system resolution and field-of-view, motion system fidelity, and simulator system delays.

So that cause-and-effect relationships may be more easily established between these variables and their consequences, single-axis compensatory tracking tasks have been used experimentally.^{1,5-14} The majority of previous compensatory tracking motion research has concentrated on the roll axis, while only a few studies have examined the vertical axis¹ or the directional axis.⁵ The motion aspect of the vertical-axis study in Ref. 1 primarily evaluated the influence of high-pass filter natural frequency on pilot-vehicle performance, with little emphasis on the effects of filter gain. The directional-axis portion of the Ref. 5 study evaluated only full motion against no motion. Thus, the purpose of this study was to examine a broader spectrum of high-pass motion

* Aerospace Engineer, Senior Member AIAA.

filter variations in the vertical and directional axes than has been examined previously.

In this experiment, the NASA Ames Vertical Motion Simulator was used with a simplified helicopter model to examine the effects of motion cue attenuation and phase error on pilot-vehicle performance and opinion. The variations in motion cue attenuation and phase error spanned the motion fidelity criteria suggested by Sinacori¹⁵ and included full motion and fixed-base evaluations at the extremes.

First, the task and the experimental apparatus are described. Then the results, which are composed of objective pilot-vehicle performance metrics and subjective motion fidelity ratings and pilot comments, are presented and discussed.

Experiment Description

Pilots performed a single-axis target-tracking task in the presence of a disturbance with the experimental variables being only gain and natural frequency of the high-pass motion filter. The vertical and directional axes were examined independently. The pilot controlled the model a single axis at a time, while positions of all the remaining axes were always zero. Therefore, in the following sections, the details of the task and the vehicle dynamics for the vertical and directional axes are discussed separately.

Tasks

Vertical. Figure 1 shows the display presented to the pilot. The object was to null the error between the moving target aircraft and the fixed horizontal dashed line. A system block diagram that depicts how the error developed is shown in Fig. 2. Two external inputs were used in a scheme similar to that developed in Ref. 8. The target was driven by a sum-of-sines (SOS) input, and the vehicle was disturbed by another SOS input that was summed with the pilot's collective position. Each of these sum-of-sines was:

$$\begin{aligned} i_h(t) = & 2.573 \sin(0.15t) + 2.202 \sin(0.34t) \\ & + 1.563 \sin(0.64t) + 0.923 \sin(1.13t) \\ & + 0.411 \sin(2.05t) + 0.150 \sin(3.56t) \\ & + 0.040 \sin(6.32t) \end{aligned} \quad (1)$$

$$\begin{aligned} \delta_{c_g}(t) = & 0.029 \sin(0.28t) + 0.058 \sin(0.49t) \\ & + 0.099 \sin(0.80t) + 0.167 \sin(1.50t) \\ & + 0.209 \sin(2.67t) + 0.201 \sin(4.63t) \\ & + 0.148 \sin(8.50t) \end{aligned} \quad (2)$$

Each component of each wave completed an

integral number of cycles in the task time length of 204.8 sec. A warm-up period of 10 sec preceded the run, and a cool-down period of 3 sec followed the run. To prevent the pilot from separating target motion from disturbance motion, the disturbance input, δ_{c_g} , was selected so that its resulting altitude spectral content (when filtered by the vehicle dynamics) matched the target shaping function.¹³ In order to adequately balance visual target motion and the cockpit accelerations and displacements, this shaping function was determined empirically. The compromised result of this empirical shaping was that the highest frequency component of the target input was below the simulator's visible threshold of 3-4 mins of arc, and the lowest component of the disturbance input was below the linear acceleration detection threshold of 0.01 g's.⁸ As shown in Fig. 2, the pilot received two external cues for use in zeroing the target error: a visual cue, e , and a motion cue, h_{sim} . The dynamics between the pilot input and these cues are discussed in the sections that follow; however, the only block in Fig. 2 that was modified in this experiment is the one labelled "motion filter." The details of these blocks are described later.

While the pilot was instructed to constantly null the displayed error, the desired performance for the task was to keep the error within one-half the height of the target vertical tail for half of the run length. The target was placed 100 ft in front of the aircraft, and the height of the vertical tail was 3 ft.

Directional. This task was analogous in structure and purpose to the vertical task. The target and a fixed vertical dashed line were oriented as shown in Fig. 3. The pilot was again instructed to null the displayed error, and the error was generated using the structure in Fig. 4. The target input shaping function was the same as in the vertical axis, with i_r replacing i_h . The target was placed 50 ft in front of the aircraft, and the target's small horizontal line subtended approximately 3 deg of visual angle. The disturbance input, which was now added to the pilot's pedal input, was adjusted to reflect the differences between the aircraft's vertical and directional dynamics, which are discussed later. The directional disturbance input was:

$$\begin{aligned} \delta_{r_g}(t) = & 0.011 \sin(0.28t) + 0.023 \sin(0.49t) \\ & + 0.039 \sin(0.80t) + 0.066 \sin(1.50t) \\ & + 0.082 \sin(2.67t) + 0.079 \sin(4.63t) \\ & + 0.058 \sin(8.50t) \end{aligned} \quad (3)$$

Vehicle Dynamics

Vertical. The model's vertical-acceleration to

collective-position dynamics are given by the following transfer function:

$$\frac{\ddot{h}}{\delta_{\text{tot}}}(s) = \frac{9s}{s + 0.3} \quad (4)$$

The collective sensitivity and the model's vertical damping were selected by averaging the hover, sea-level characteristics given in Ref. 16 of five helicopters: the OH-6A, BO-105, AH-1G, UH-1H, and the CH-53D. While a delay is usually added to this transfer function to approximate the lag due to the vehicle's higher-order dynamics, this delay was instead subsumed by the hardware delays in the simulator motion and visual systems. This technique was successful in a previous experiment.¹⁷ In addition, no torque or RPM limits were imposed on the pilot. Also, since the task was single-axis, no response coupling from the collective input into the directional axis occurred.

Directional. The directional-axis dynamics were selected to be a low-order representation of a CH-53D in hover, which are approximately¹⁶

$$\frac{\ddot{\psi}}{\delta_{\text{rot}}}(s) = \frac{22.9s}{(s + 0.3)} \quad (5)$$

The response was linear, and no vortex-wake or power-limiting effects were modeled. Also, to examine only the effects of yaw rotational cueing on the pilot, no side force cues from pedal inputs were represented. Thus, for experimental purposes, a pedal input produced only a yaw couple, and the pilot was intentionally placed near the aircraft's center of rotation.

Simulator and Cockpit

The NASA Ames Vertical Motion Simulator (VMS) was used. The mainframe computer cycle time was 25 msec. A cutaway view of the motion system with its nominal software-command position, velocity, and acceleration limits is shown in Fig. 5. Again, throughout the experiment, the displacements in the lateral, longitudinal, pitch, and roll axes were always zero. Within the listed limits, the dynamic performance of the simulator depends on the axis. Using frequency response testing techniques,¹⁸ the dynamic characteristics of the vertical and directional axes were determined. Over the handling qualities frequency range of interest of 0.1-10 rad/sec, accurate models are given in the linear range by:

$$\frac{\ddot{h}_{\text{sim}}}{\ddot{h}_{\text{filt}}}(s) = \frac{(8)(26)}{(s + 8)(s + 26)} \quad (6)$$

$$\frac{\ddot{\psi}_{\text{sim}}}{\ddot{\psi}_{\text{filt}}}(s) = \frac{11^2}{s^2 + 2(0.6)(11)s + 11^2} \quad (7)$$

These dynamics represent those in the "motion hardware" blocks of Figs. 2 and 4.

The Singer Link DIG-I visual system was used, which has visual delay from the math model to the generation of a visual image of 0.083 sec.¹⁹ This delay characterizes the block labelled "visual hardware" in Figs. 2 and 4.

To approximately compare the delays between the visual and motion cues, the phase angle responses of eqns. (6) and (7) were each matched with a pure time delay. This matching resulted in the vertical and directional motion axes having 0.14 sec and 0.13 sec of delay, respectively. Thus, for these two axes in the VMS, the visual response effectively leads the motion response.

The visual field-of-view was presented on a single window that spanned +/-22 degs horizontally and +/-12 and -16 degs vertically. Only the target aircraft was presented in the scene. The remainder of the scene was completely obscured by fog.

A conventional left-hand collective and pedals were used. The collective had an available travel of +/- 5 in, had no force gradient, and the friction was adjustable by the pilot. The pedals had a travel of +/- 2.7 in, a breakout force of 3.0 lb, a force gradient of 3 lb/in, and a damping ratio of 0.5.

All flight instruments were disabled making the visual scene and motion system the only cues available to the pilot. Rotor and transmission noises were present to partially mask the motion system noise. Six NASA Ames test pilots participated, hereafter referred to as pilots A-F. All pilots had extensive rotorcraft flight and simulation experience. The pilots sat 0.5 ft forward of the yaw axis, and they were located on the aircraft centerline. For this position, negligible longitudinal and lateral linear accelerations were present.

Motion Filter Configurations

Second-order motion filters of identical form were used for both the vertical and directional axes. The motion filters had the form (Figs. 2 and 4):

$$\frac{\ddot{h}_{\text{filt}}}{\ddot{h}}(s) = \frac{\ddot{\psi}_{\text{filt}}}{\ddot{\psi}}(s) = \frac{Ks^2}{s^2 + 2\zeta\omega_m s + \omega_m^2} \quad (8)$$

The damping ratio (ζ) was 0.7 for all configurations; only the gain (K) and the natural frequency (ω_m) were varied. Table 1 lists K and ω_m for each tested configuration in each axis.

The values in Table 1 were selected to span the motion fidelity criteria developed in Ref. 15,

which are shown in Figs. 6 and 7 for the vertical and directional axes. Each configuration is located on either Fig. 6 or Fig. 7 by substituting K and ω_m into eqn. (8) and evaluating the filter's magnitude and phase angle at 1 rad/sec, which was selected as the evaluation frequency in Ref. 15. Phase distortion is defined as the phase angle at 1 rad/sec, since the phase angle for perfect motion would be zero. Of course, since the filter is high-pass, the phase distortion is in the form of lead.

The criteria have three motion fidelity levels: high, medium, and low. High fidelity means "motion sensations are close to those of visual flight." Medium fidelity means "motion sensation differences are noticeable but not objectionable." Low fidelity is defined as "differences are noticeable and objectionable, loss of performance, disorientation."¹⁵ As expected, high motion fidelity is associated with high motion gain and low phase distortion, and low motion fidelity is associated with low motion gain and high phase distortion.

Procedure

All configurations were tested blind and were randomized. Pilots were asked to rate the motion fidelity of each configuration, using the motion fidelity definitions given previously. Between each configuration, in order to calibrate or recalibrate themselves on the true response of the vehicle model, pilots flew the model with full motion (configuration V1 or D1) in a visual scene against objects of known size. Further details of these objects are given in Ref. 4. All six pilots flew all ten of the vertical configurations. On average, three pilots flew each of the seven directional configurations.

Results

Vertical

Time histories and standard deviations of several pertinent variables for a full motion case (V1) and a no motion case (V10) are shown in Figs. 8 and 9. Both of these runs were made by the same pilot. A comparison of these two figures reveals that when going from full motion to no motion, the target error, vehicle acceleration, and collective displacement all increase. Rather than compare time histories across the ten cases and six pilots, several pilot-vehicle performance metrics were determined, and their statistical significance was evaluated.

First, since the two sum-of-sines inputs were statistically independent, two effective open-loop pilot-vehicle describing functions may be determined.¹³ One open-loop describing function applies to the target errors due to target motion; it is given by calculating the ratios of the Fourier

coefficients of $h(j\omega)/e(j\omega)$ at the target input frequencies. The other describing function applies to target errors due to the disturbance input, and it is determined from the ratios of the Fourier coefficients of $-\delta_c(j\omega)/\delta_{c10}(j\omega)$ at the disturbance input frequencies. These two describing functions are referred to as "target following" and "disturbance rejection" hereafter.

From these describing functions, open-loop crossover frequencies and phase margins were evaluated. The crossover frequency effectively indicates how fast the error is initially zeroed. The higher the crossover, the faster the initial nulling of the error will be. The phase margin effectively indicates the damping of the error response. The higher the phase margin, the more well damped the error response will be.

Each of these open-loop describing functions includes a different combination of both the pilot's internal visual and motion compensation applied to the visual error and the acceleration feedback.^{8,13} The identification of these compensations is not presented in this paper. While only the motion cues were changed here, the pilot's internal compensation may change in an attempt to account for degradations in either the visual or motion cues. Instead of examining these changes, the overall effect of these changes on pilot-vehicle performance and opinion are given.

Target Following. The target-following, pilot-vehicle, open-loop crossover frequencies are given in Fig. 10 for all of the configurations in descending order of mean value. For easy reference, the motion filter gain and natural frequency, rounded to one decimal place, are indicated above each bar. The mean value (shaded bar) and standard deviation (clear portion above shaded bar) are shown, each of which is determined from six values. Each of the six values for determining the mean corresponds to the individual average for each pilot. Upon examining the variance ratio,²⁰ or F-test, for these data, the differences among the configurations were not significant at the 5% level. In addition, no coherent trend is present among the results in view of the physical variations made in the motion filter. This result agrees with, and extends, the data in Ref. 1, which indicates an invariance in target-following open-loop crossover frequency for $0.2 < \omega_m < 1.25$ rad/sec with $K=1$. Reference 1 also shows an invariance for the two values of vehicle vertical-velocity damping that were tested.

Figure 11 shows the target-following loop phase margins for all configurations. Unlike the crossover results, true differences are apparent in the phase margins, although the range of the means is only 10 degs. The largest phase margin occurred with the full motion condition (V1), and phase margin was progressively lost as both K was reduced and

ω_m was increased. Using the Newman-Keuls method²⁰ to determine which means are statistically different at the 5% level, the results indicate that configurations V1 and V3 were different from V4 and V6-V10, and configurations V2 and V5 were different from V8.

These results are partially consistent with those of Ref. 1, which indicated that target tracking phase margin degraded as ω_m was increased from 0.2 to 1.25 rad/sec at $K=1$. However, Ref. 1 exhibited larger variations in the measured phase margins, including an almost 20 deg variation between $\omega_m=0.2$ and 0.5 rad/sec. Only a slight difference was measured in this experiment between the nearly equivalent V1 and V2 motion configurations.

The overall vertical target following results suggest that the speed at which the target is initially nulled appears to be unaffected by variations in motion; however, the resulting damping of the error to a nulled steady-state value is slightly affected by motion variations.

Reference 4, which examined a different target-following task with the same vehicle dynamics, suggests that the high fidelity portion of the criterion in Ref. 15 should include configuration V5. The performance results shown here for target following are consistent with that suggestion.

Disturbance Rejection. Pilot-vehicle open-loop crossover frequencies for the disturbance-rejection loop are given in Fig. 12. The crossover magnitudes appear to be roughly ordered by phase distortion level, that is, successive increases in ω_m . The statistical results reveal that, at the 5% level, configurations V2, V1, and V7 all had higher crossover frequencies than the fixed base case V10. Again, these results are consistent with the configurations tested in Ref. 1, and they extend the results by suggesting that the open-loop crossover appears to be affected by changes in ω_m at all levels of K .

Figure 13 shows the disturbance-rejection loop phase margins. Here, the configurations seem to be ordered by progressive reductions in K . Statistically, configuration V3 was better than all of the other cases, and configurations V1 and V4 were better than V10. Ref. 1, for $K=1$, also shows that the low and high phase distortion cases have roughly the same phase margin, with perhaps a slight peaking at moderate phase distortion. Here, more relative peaking in phase margin was observed for the V3 case than given in Ref. 1. The crossover frequency for V3 was reduced, which alone might contribute to an increased phase margin; however, V4 had a low crossover, but not an accompanying phase margin peak. With the unknown pilot feedbacks, the V3 configuration must be coupling in with the vehicle dynamics in a manner different from the

other configurations.

Total Tracking Error. Vertical tracking errors are shown in Fig. 14. This error accrues from both the target input and the disturbance input. The four lowest errors occurred for the lowest phase error configurations, but the differences between any two of the motion configurations were not large. The Newman-Keuls results indicated that all of the motion configurations, V1-V9, had better performance than the no motion case V10, while no tracking error differences were present among the V1-V9 cases at the 5% level.

Pilot Opinion. Figure 15 gives the motion fidelity ratings using the definitions stated earlier. The ratings are divided into high, medium, low, and split, where "split" refers to a pilot assigning inconsistent ratings for repeat runs of a given configuration. Split ratings only occurred for the V1 configuration, which was more likely, since it had more repeat examinations than the other cases. The ratings indicate that no pilot perceived the $K=0.3$ cases to be high fidelity. All pilots rated the V10 case to be low fidelity. Configuration V2 received the best overall ratings (which also had the lowest mean tracking error, as shown earlier). Configuration V3 surprisingly received two low ratings, while V4 did not receive any low ratings. Configuration V6, which is essentially a combination of V3's natural frequency and V5's gain, was rated worse than V3 or V5.

Directional

Figures 16 and 17 depict key variables for a full motion case (D1) and a no motion case (D7). The maximum yaw rates for Figs. 16 and 17 (not shown) were 9.1 and 9.8 deg/sec, respectively. The yaw-rate standard deviations were 2.3 and 2.6 deg/sec, respectively. Unlike the vertical axis, the differences between these two extremes are not apparent. The tracking errors, aircraft movements, and cockpit control displacements are very similar. Pure yaw motion appears to have little effect on the pilot-vehicle performance for this task.

Target Following. Pilot-vehicle open-loop crossover frequencies and phase margins are shown in Figs. 18 and 19, respectively. No statistical differences at the 5% level were noted for either metric. Only an average of three pilots flew each of these configurations, which requires that more severe differences must be present to be confident of actual inter-configuration differences than with the sample size for the vertical tests. However, sample size may not be an issue, since the means in Figs. 18 and 19 do not exhibit any coherent trends for the associated motion filter changes. For instance, the full motion configuration, D1, had the lowest mean crossover frequency, while the no motion case, D7, had the third best. It appears that the pure yaw motion did not have any effect on target following.

Disturbance Rejection. Figures 20 and 21 show the open-loop pilot-vehicle crossover frequencies for the yaw disturbance loop. Again, no statistical differences were present in either metric at the 5% level. That result is evident from the presented means and standard deviations. The disturbance loop results also indicate that pure yaw motion had no effect on the pilot-vehicle performance.

Total Tracking Error. Yaw tracking errors for each configuration are given in Fig. 22. No statistical differences were present at the 5% level. Here, the no-motion mean performance was better than the full-motion mean performance, which suggests that any of the differences among configurations are likely due to random sampling.

Pilot Opinion. Finally, Fig. 23 gives the pilot motion fidelity ratings, as defined earlier, for the task. From the ratings, pilots apparently were able to distinguish when the cab was and was not moving, although this was not always the case. For the full motion case, D1, pilot comments were that the configuration felt like the full motion case, but that the full motion feeling was far from compelling. Also, pilots noted that whatever configuration changes were being made, they were having no effect on the task execution or performance.

The above directional results differ with those of Ref. 5. The Ref. 5 directional results indicate that 100 msec of delay in the pilot-vehicle describing function were eliminated when going from fixed-base to full motion in a disturbance rejection task. Also, the tracking error was reduced between these two tested configurations. Several differences exist between this experiment's set up and that of Ref. 5. The latter had only an integrator between the pilot input and the vehicle's yaw angle, which is indicative of a very high level of yaw damping. Thus, larger angular accelerations would result from high frequency pilot inputs than would occur herein. In addition, the Ref. 5 disturbance input rms was 15 deg. In this experiment, the target input had an rms of 2.7 degs and a disturbance rms of 2.3 degs. Thus the level of motion cues in the Ref. 5 experiment was higher than that reported here.

The results from Ref. 4, which were for a target following task with the same yaw dynamics, also revealed that yaw cues had little-to-no influence on task performance or pilot opinion.

Conclusions

This experiment, conducted on the NASA Ames Vertical Motion Simulator, evaluated how changes in a filter that commands motion of a flight simulator cockpit affect pilot-vehicle performance and pilot opinion of motion fidelity. A single-axis

compensatory tracking task was used for evaluation in either the vertical or the directional axis. Two inputs, a randomly moving target and a simulated disturbance, were used so that the motion effects on both target following and disturbance rejection could be evaluated. Specifically, the conclusions were:

1. For vertical target following, pilot-vehicle crossover frequency was not affected by the motion configurations. This result is consistent with, and extends, previous results.
2. Pilot-vehicle phase margins for vertical target following improved as either the motion gain increased or as natural frequency decreased. This result is generally consistent with previous data; however, smaller variations in phase margins were measured in this experiment.
3. Vertical disturbance rejection crossover frequencies improved with decreasing motion filter natural frequency, which is consistent with, and extends, earlier data.
4. Vertical disturbance rejection phase margins exhibited a peak at a moderate natural frequency (0.5 rad/sec). This peak was larger than that experienced in a previous experiment.
5. Total vertical tracking error significantly increased when vertical motion was completely removed.
6. No significant differences resulted among the directional configurations in which the only cue to the pilot was pure rotation about the vertical axis. Pilots had difficulty in detecting the presence of yaw motion during the task. Further work should examine other tasks to determine the effectiveness of yaw motion cues in flight simulation, since yaw motion was not helpful for the task in this experiment.

References

- ¹Bray, R.S., "Visual and Motion Cueing in Helicopter Simulation," NASA TM 86818, Sept. 1985.
- ²Rolle, J.M. and Staples, K.J., *Flight Simulation*, Cambridge University Press, Cambridge, 1986, p.117.
- ³Baarspul, M., "A Review of Flight Simulation Techniques," *Progress in Aerospace Science*, Vol. 27, 1990, pp. 1-120.
- ⁴Schroeder, J. A., "Evaluation of Simulation Fidelity Criteria in the Vertical and Directional Axes," American Helicopter Society 49th Annual Forum, St.

Louis, MO, May, 1993.

⁵Meiry, J.L., "The Vestibular System and Human Dynamic Space Orientation," NASA CR-628, 1966.

⁶Young, L.R., "Some Effects of Motion Cues on Manual Tracking," *AIAA Journal of Spacecraft*, Vol. 4, No. 10, 1967, pp. 1300-1303.

⁷Shirley, R.S. and Young, L.R., "Motion Cues in Man-Vehicle Control," *IEEE Transactions on Man-Machine Systems*, Vol. MMS-9, No. 4, 1968, pp. 121-128.

⁸Stapleford, R.J., Peters, R.A., and Alex, F.R., "Experiments and a Model for Pilot Dynamics With Visual and Motion Inputs," NASA CR-1325, 1969.

⁹Bergeron, H.P., "The Effects of Motion Cues on Compensatory Tasks," AIAA Paper 70-352, AIAA Visual and Motion Simulation Technologies Conference, 1970.

¹⁰Sinacori, J.B., "A Practical Approach to Motion Simulation," AIAA 73-931, AIAA Visual and Motion Simulation Technologies Conference, Palo Alto, CA, 1973.

¹¹Junker, A.M. and Price, D., "Comparison Between a Peripheral Display and Motion Information on Human Tracking About the Roll Axis," AIAA Visual and Motion Simulation Conference, Dayton, OH, 1976.

¹²van Gool, M.F.C., "Influence of Motion Washout Filters on Pilot Tracking Performance," AGARD-CP-249, 1978.

¹³Jex, H.R., Magdaleno, R.E., and Junker, A.M., "Roll Tracking Effects of G-Vector Tilt and Various Types of Motion Washout," NASA CP-2060, 1978, pp. 463-502.

¹⁴van der Vaart, J.C. and Hosman, R.J.A.W., "Compensatory Tracking in Disturbance Tasks and Target Following Tasks. The Influence of Cockpit Motion on Performance and Control Behaviour," Report LR-511, Delft University of Technology, The Netherlands, Dec. 1987.

¹⁵Sinacori, J.B., "The Determination of Some Requirements for a Helicopter Flight Research Simulation Facility," NASA CR-152066, Sept., 1977.

¹⁶Heffley, R.K, Jewell, W.F., Lehman, J.M., and Van Winkle, R.A., "A Compilation and Analysis of Helicopter Handling Qualities Data. Volume One:

Data Compilation," Technical Report No. 1087-1, Systems Technology, Inc., 1979.

¹⁷Schroeder, J.A., Tischler, M.B., Watson, D.C., and Eshow, M.M., "Identification and Simulation Evaluation of an AH-64 Helicopter Hover Math Model," AIAA-91-2877, AIAA Atmospheric Flight Mechanics Conference, New Orleans, LA, Aug. 1991.

¹⁸Tischler, M.B. and Cauffman, M.G., "Frequency-Response Method for Rotorcraft System Identification with Applications to the BO-105 Helicopter," American Helicopter Society 46th Annual Forum, Washington, D.C., May, 1990.

¹⁹McFarland, R.E., "Transport Delay Compensation for Computer-Generated Imagery Systems," NASA TM 100084, Jan. 1988.

²⁰Snedecor, G.W. and Cochran, W.G., *Statistical Methods*, Sixth Edition, The Iowa State University Press, Ames, Iowa, 1967, p. 265.

Table 1 Motion Filter Configurations

Configuration	Vertical	
	K	ω (rad/sec)
V1	1.000	0.010
V2	0.901	0.245
V3	1.000	0.521
V4	1.000	0.885
V5	0.650	0.245
V6	0.670	0.521
V7	0.300	0.245
V8	0.309	0.521
V9	0.377	0.885
V10	0.000	--
Configuration	Directional	
	K	ω (rad/sec)
D1	1.000	0.010
D2	1.000	0.521
D3	1.000	0.885
D4	0.702	0.245
D5	0.312	0.521
D6	0.128	0.885
D7	0.000	--

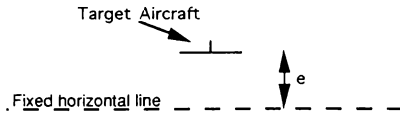


Figure 1 - Pilot's display for vertical task

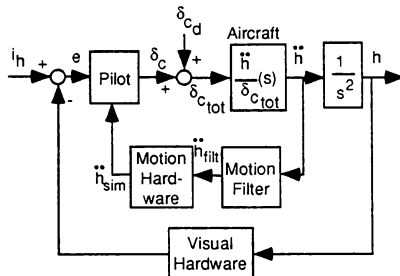


Figure 2 - Vertical compensatory loop

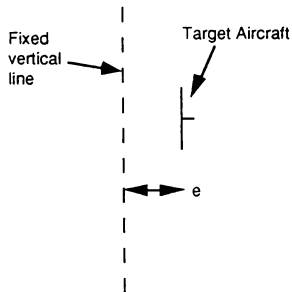


Figure 3 - Pilot's display for directional task

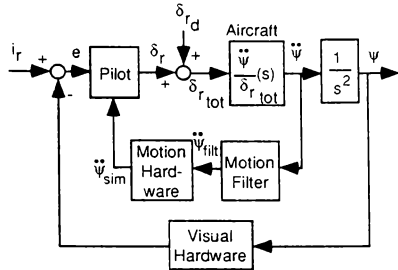


Figure 4 - Directional compensatory loop

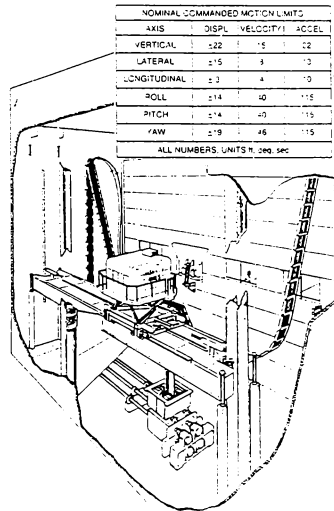


Figure 5 - Vertical Motion Simulator

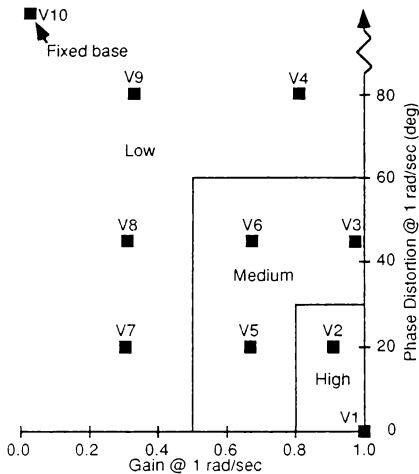


Figure 6 - Vertical configurations on Ref. 15 specific force criterion.

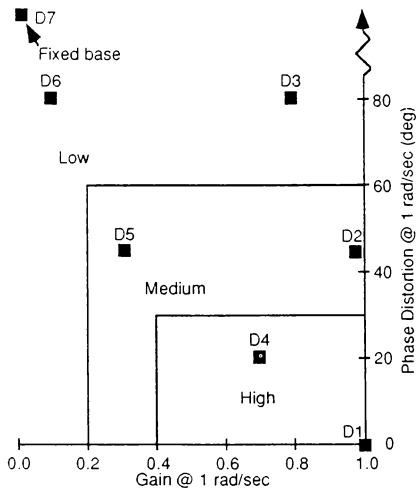


Figure 7 - Directional configurations on Ref. 15 rotational velocity criterion.

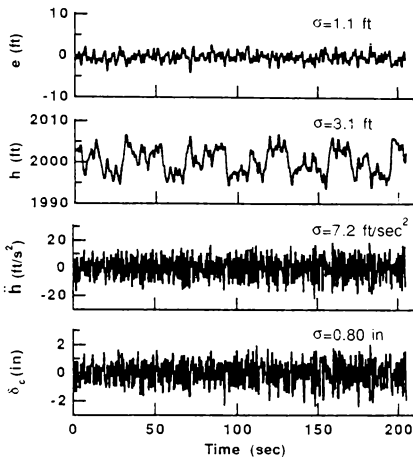


Figure 8 - Example time history, full motion (V1), pilot C.

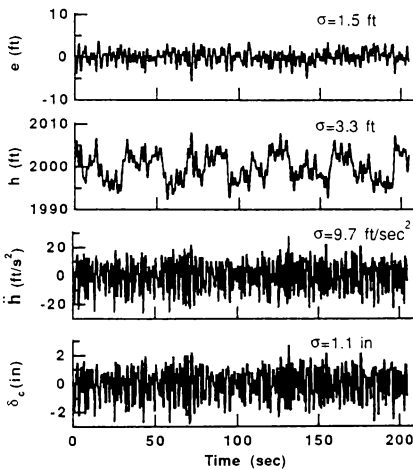


Figure 9 - Example time history, no motion (V10), pilot C.

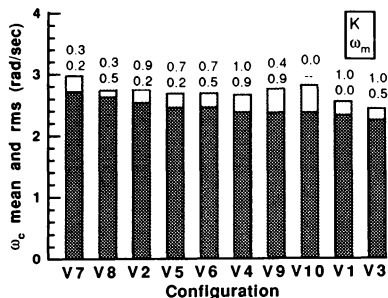


Figure 10 - Vertical target-following pilot-vehicle open-loop crossover frequencies.

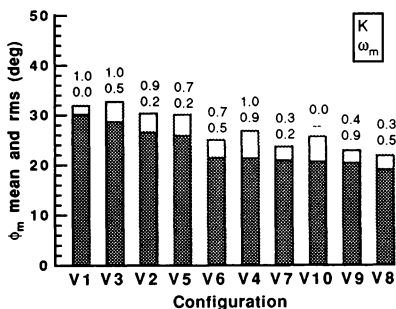


Figure 11 - Vertical target-following pilot-vehicle open-loop phase margins.

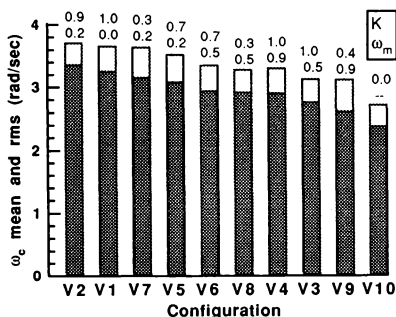


Figure 12 - Vertical disturbance-rejection pilot-vehicle open-loop crossover frequencies.

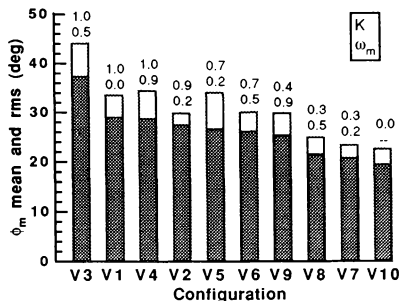


Figure 13 - Vertical disturbance-rejection pilot-vehicle open-loop phase margins.

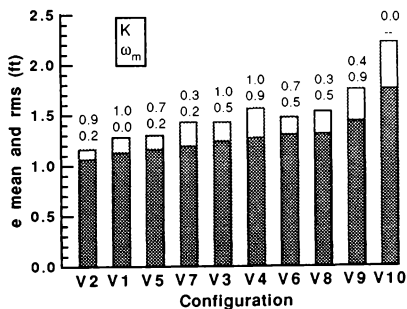


Figure 14 - Vertical tracking errors.

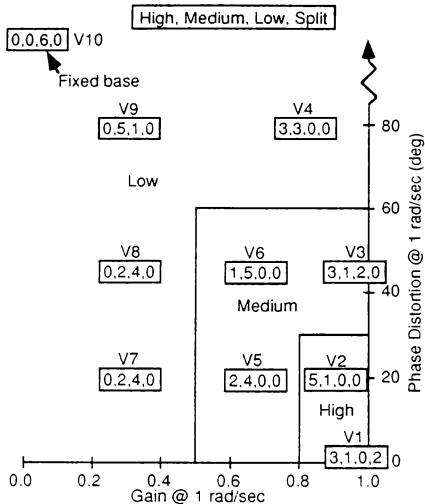


Figure 15 - Pilot motion fidelity ratings for vertical task.

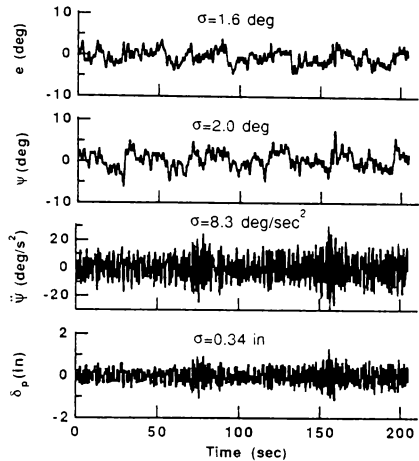


Figure 17 - Example time history, no motion (D7), pilot D.

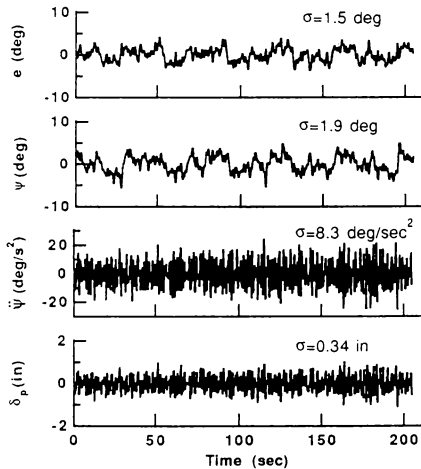


Figure 16 - Example time history, full motion (D1), pilot D.

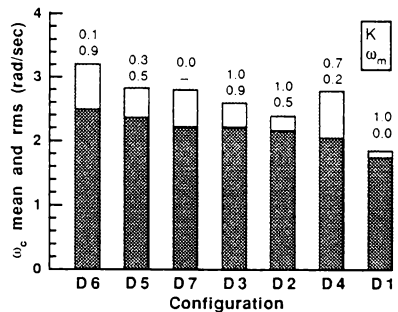


Figure 18 - Directional target-following pilot-vehicle open-loop crossover frequencies.

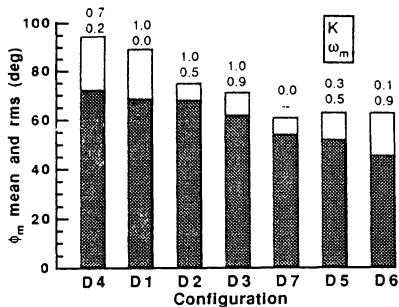


Figure 19 - Directional target-following pilot-vehicle phase margins.

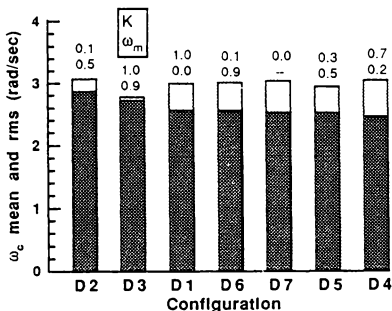


Figure 20 - Directional disturbance-rejection pilot-vehicle open-loop crossover frequencies.

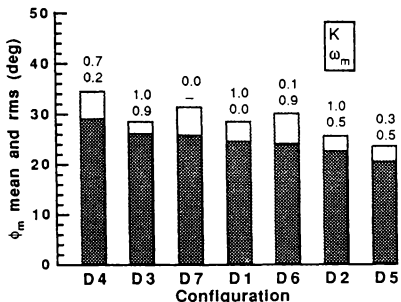


Figure 21 - Directional disturbance-rejection pilot-vehicle open-loop phase margins.

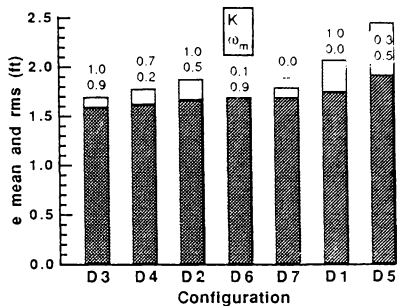


Figure 22 - Directional tracking errors.

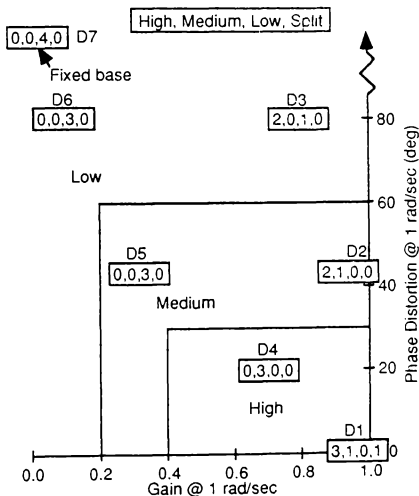


Figure 23 - Pilot motion fidelity ratings for directional task.

PILOT EVALUATIONS OF AUGMENTED FLIGHT SIMULATOR MOTION

Stuart T. Garrood* and Lloyd D. Reid**

University of Toronto Institute for Aerospace Studies
4925 Dufferin Street
Downsview, Ontario, Canada, M3H 5T6

Abstract

In an earlier paper it was proposed that the turbulence induced motion of a flight simulator could be augmented without affecting the visual and instrument displays. This may be necessary if the simulator's washout filters severely restrict its motion response to atmospheric turbulence. The present study has implemented the proposed technique on the University of Toronto Institute for Aerospace Studies Flight Research Simulator and carried out pilot evaluations for both high altitude and low altitude operations in the presence of atmospheric turbulence. The results indicate that the technique can successfully increase the simulator's motion response to turbulence in a manner that is acceptable to pilots. It was found that a simple second-order transfer function representation of the aircraft is sufficient within the motion augmentation channel. The resulting motions were judged to add to the realism of the simulation and to compare favourably with other training simulators.

Introduction

In Reference 1 a technique for augmenting flight simulator motion response to turbulence was proposed. The present paper describes how this technique was implemented on a flight simulator and subjected to pilot evaluations.

The need for augmenting the motion response due to turbulence arises because the software (washout filters) used to drive the motion system is designed to produce realistic motion cues while restricting the resulting motion to remain within the physical limitations imposed by the ground-based motion hardware. When these motion-drive washout algorithms are tuned to accommodate pilot-induced motions they often have the undesired side effect of overly attenuating the heave response of the simulator to turbulence. It is shown in References 1 and 2 that one need only consider the aircraft's heave response to the vertical component of

*Graduate Student

**Professor and Associate Director, Associate Fellow AIAA

turbulence in order to compensate for this effect. The scheme devised to accomplish this is illustrated in Figure 1. The input to the flight equations of block 1 are the pilot's controls δ , and the Turbulence T_1, T_2 (as shown in Reference 1) can be taken to be a scaled version of W_G , the vertical component of turbulence. The flight equations of block 2 are represented by a simplified transfer function relating the aircraft's heave acceleration response (\ddot{a}_B^Z) to W_G . \ddot{a}_B^Z is used to augment the motion-drive signal coming from block 1 without disturbing the visual display or the instruments.

The goals of the present study were to:

- (1) Confirm that augmenting the simulator motion will not result in exceeding the travel limits of the motion hardware when there is a pilot in the loop.
- (2) Determine whether the proposed method of augmenting the simulator motion produces motion cues acceptable to pilots.
- (3) Obtain pilot evaluations of turbulence induced motion cues over a range of conditions when employing the augmentation technique.

Aircraft Transfer Functions

The aircraft employed in the present tests was a B-747 implemented on the UTIAS Flight Research Simulator (see Figure 2). It was decided to evaluate the motion augmentation scheme for the aircraft in both a cruise configuration and in a landing configuration. For cruise the aircraft was taken to be clean and in level flight at 295 kt, indicated airspeed (KIAS) at an altitude of 21,200 ft. For landing the aircraft was evaluated with 25° flaps and the landing gear down and flown in level flight at 170 KIAS at an altitude of 2,500 ft.

Describing functions relating \ddot{a}_B^Z to W_G were determined by using power spectral density measurements based on input (W_G) and output (\ddot{a}_B^Z) data produced while flying through the complete turbulence field. Bode plots were generated of the form

$$DF(\omega) = \frac{\Phi_{W_G u_R^*}(\omega)}{\Phi_{W_G W_G}(\omega)} \quad (1)$$

The resulting plots are shown in Figures 3 and 4. As described in Reference 2, low-order transfer functions were fitted to these plots. For the cruise case two second-order transfer functions F1 and F2 were generated. As shown in Figure 3, F2 produced a much better fit to the data, particularly at the higher frequencies. For the landing case two second-order transfer functions L1 and L2 and a fourth-order transfer function L3 were generated. As shown in Figure 4, L3 produced the best fit and L1 the worst. However, the increased complexity of the better transfer functions must be paid for by the extra time required to select their parameters and the computer overhead associated with their implementation in real time on the simulator.

Flight Conditions

The turbulence model employed was the M1 version developed in Reference 3 which was judged to be the most realistic in pilot evaluations of several alternatives. It is based on filtered white noise modified to produce non-Gaussian turbulence. The turbulence model produces three uncorrelated gust velocities in the aircraft body-fixed reference frame. These are applied at the aircraft's center of gravity. Rolling and yawing gusts are generated and are combined with the aircraft rotational rates to represent the influence of spatial gust gradients. A gust pitching effect due to the horizontal tail is included by adjusting the tail angle-of-attack in response to a time delayed version of W_G .

Two flying tasks were employed when the evaluation pilots were assessing the turbulence induced motion. The first was a loose formation flight in which the lead aircraft flew a racetrack pattern² as shown in Figure 5. Turbulence was present during the complete run. The height for the flight was 20,700 ft. AGL (with ground level being at 500 ft. altitude) and the airspeed was 295 KIAS. The second task was the approach to landing depicted in Figure 6. As shown in the figure, turbulence was present from part way through Section 2 until that point in Section 4 where the height goes below 500 ft. AGL. Section 1 begins with the aircraft at 6,000 ft. AGL and 220 KIAS. The objective during this section is to decelerate to 170 KIAS and deploy 25° flaps and landing gear. In Section 2 the aircraft descends to a height of 2,000 ft. AGL. Following a sidestep maneuver in Section 3 an ILS approach is made in Section 4.

Pilots

The five pilots participating in this study were either active airline pilots or active company pilots. All of the subjects took part on a voluntary basis. While the pilots were not previously experienced on the B-747, all were familiar with transport aircraft and had significant simulator experience. Table 1 outlines their flying and simulator hours.

Experimental Plan

The turbulence levels for the flying tasks were selected so that the turbulence augmentation technique generated a significant increase in heave specific force without causing the simulator to reach its physical travel limits. It was found that W_G for the landing task could be scaled up by a factor of 2 relative to that employed for the formation flight while still meeting the travel limits requirement. Typical plots of the standard deviation of vertical specific force in the simulator are shown in Figure 7. It can be seen that the augmented cases produced approximately twice the non-augmented levels.

Data were gathered for both complete single 10 min. executions of the flight paths depicted in Figures 5 and 6 and paired comparison trials were carried out using the indicated section in Figure 5 and Sections 3 and 4 of Figure 6. Each of the latter flights lasted 4 min. Each of the above flights was flown without turbulence augmentation (NA) and with F1 and F2 augmentation for the formation task and with L1, L2 and L3 augmentation for the landing task.

Each pilot participated in 2 days of training and data gathering. The formation task was flown on the first day and the landing task on the second. Each day began with task training using the full flight paths. They were allowed to continue to practice until they felt comfortable with the task. This was followed by data gathering employing the full flight paths first and then the paired comparison tests using the shorter flights. Within each of these blocks each pilot tested all the turbulence treatment cases once in a randomized sequence.

The pilots answered a questionnaire at the end of each 10 min. run and a number of performance and motion measures were made. The following questions from the questionnaire were directed towards the goals of the present study and are reported herein:

- (1) Did the turbulence add to the realism of the simulation?

- (2) Did the heave motion response to turbulence compare favourably to that experienced in heavy transport aircraft?
- (3) Did the heave motion response to turbulence compare favourably to that experienced in training simulators?

The object of the paired comparison tests was to obtain a ranking of the augmentation formulations. The pilots flew the same maneuver twice in a row with a different pair of turbulence treatments (including NA) in each set (3 pairs for the formation case and 6 pairs for the landing case). The ordering of the pairs was randomized and each pair was flown once by each pilot. Within each pair the pilot must indicate which turbulence treatment he felt best simulated the aircraft's response to turbulence. A ranking must be given in all cases, even if the treatments appear to be very similar. This generates a hierarchical ordering of the treatments, the validity of which can be checked by analyzing the internal consistency of each pilot's rankings, and the consistency among the pilots.

Experimental Results

An analysis of variance was carried out on the performance measures and it was determined that pilot performance was not influenced by the turbulence motion augmentation employed in the present study. For this reason details of this aspect of the study are not included here. They can be found in Reference 2.

Formation Flight

The responses to the three questions concerning the realism of the three cases NA, F1 and F2 are summarized in Figure 8. The pilots' responses to the first question indicate that all three cases were felt to add to the realism of the simulation. Only one pilot out of the five disagreed with this for F1 and F2. This overall acceptance of F1 and F2 indicates that the proposed method for augmenting turbulence induced motion does not detract from the realism of the simulation. The responses to the second question were less encouraging. It appears that F1 and F2 induced heave motion was judged to be comparable to that experienced in heavy transport aircraft in only half of the pilots' responses. Surprisingly NA appears to have the best pilot acceptance in this instance. In the case of the third question one pilot was unable to respond due to lack of experience in simulators and thus the results are based on the remaining four pilots. As with the first question it appears that all cases compare favourably with other training simulators with only one pilot disagreeing in the cases of F1 and F2.

The paired comparison results of Table 2 indicate that four of the five pilots produced consistent results. The most consistent finding being that F2 was preferred over F1. Also shown in Table 2 is the within-judge coefficient of consistency k . If a pilot's results are perfectly consistent then k will be 1.0. If, on the other hand, his judgements exhibit the maximum inconsistency then k will be 0. In Figure 9 the analysis of the paired comparison data following the procedures outlined in Reference 4 ranks the three cases along a linear scale from worst to best. The order has F2 ranked as best and F1 as worst. In this instance the interjudge consistency is indicated by the coefficient of concordance W . The value of W will lie between 0 and 1, with 1 representing complete interjudge consistency. Here W is 0.28 and this is similar to levels found in other simulator studies.⁵

Landing Approach

The responses to the three questions concerning realism are contained in Figure 8. The pilots' responses to the first question were a solid yes for all cases. They unanimously felt that all four cases added to the realism of the simulation. As with the formation flight, for the second question, except for the L3 case, approximately half the pilot responses indicated that the induced heave motion was comparable to that experienced in heavy transport aircraft. The responses to the third question indicate that all cases compare favourably with other training simulators with only one pilot disagreeing in the cases of NA and L3.

The paired comparison results are shown in Table 3 and Figure 9. It can be seen from Table 3 that two of the pilots produced inconsistent results, both with $k = 0.2$. The ranking from Figure 9 in decreasing order of preference is L1, (L2, NA), L3. However with a coefficient of concordance of $W = 0.17$ it would appear that the order of preference is highly dependent upon which pilot you interrogate. This is reflected by the close grouping of the ranking numerical values on the linear scale when compared with the wider separation found for the formation flight results.

Summary

The turbulence induced motion augmentation technique proposed in Reference 1 has been tested by a group of five evaluation pilots flying the UTIAS Flight Research Simulator configured as a B-747. Two distinct flight regimes were employed: a high altitude formation task and a low altitude landing approach. In each instance several augmentation versions were evaluated.

The results of the study indicate that it is possible to augment turbulence induced simulator motion and achieve a significant increase in the heave specific force levels without exceeding the travel limits of the motion system when a pilot is in the loop. In the present instance this was accomplished even though the process started with a turbulence induced motion case judged to be already realistic. It was found that the augmentation process produced turbulence induced motion judged to be realistic and to compare favourably to other training simulators.

The paired comparison tests ranked the various transfer function cases from best to worst based on how well they represented the simulated aircraft's response to turbulence. For high altitude formation flight the order was F2, NA, F1 and for a low altitude landing approach the order was L1, (L2, NA), L3. Thus in both instances second-order transfer functions were judged to be best. Surprisingly NA was judged to be superior to F1 and L3.

Only NA for the formation flight and L3 for the landing approach were judged to compare favourably with turbulence response in heavy transport aircraft. This may reflect on the overall simulation environment and the difficulty of relating the present simulator motion to past infrequent experiences in actual aircraft.

On balance it appears that if a simulator is given poor pilot ratings due to insufficient turbulence induced motion then the proposed motion augmentation technique could be successfully used to correct the problem.

Acknowledgement

This project was supported by research grants provided by the Natural Sciences and Engineering Research Council of Canada.

References

1. Reid, L. D., and Robinson, P. A., "Augmenting Flight Simulator Motion Response to Turbulence," *Journal of Aircraft*, Vol. 27, No. 4, 1990, pp. 306-311.
2. Garrod, S. T., "The Refinement and Testing of a Method for the Augmentation of Flight Simulator Motion Due to Turbulence," M.A.Sc. Thesis, University of Toronto Institute for Aerospace Studies, June 1992.
3. Robinson, P. A., and Reid, L. D., "Modeling of Turbulence and Downbursts for Flight Simulators," *Journal of Aircraft*, Vol. 27, No. 8, 1990, pp. 700-707.
4. Seaver, D. A., and Stillwell, W. G., "Procedures for Using Expert Judgement to Estimate Human Error

Probabilities in Nuclear Power Plant Operations," U.S. Nuclear Regulatory Commission, NUREG/CR-2743, 1983, pp. A-1 to A-12.

5. Reid, L. D., and Nahon, M. A., "Flight Simulation Motion-Base Drive Algorithms: Part 3 - Pilot Evaluations," University of Toronto, Canada, UTIAS Report No. 319, Dec. 1986.

Table 1 Pilot experience summary

Pilot	Current position	Flying hours	Transport hours	Simulator hours
1	F/O DC-9, 767	13,500	8,000	200
2	Pilot, Dash 8	13,000	7,500	100
3	Pilot, Dash 8	12,600	12,000	55
4	Capt. A320, 767	10,000	9,000	500
5	Pilot, Dash 8	16,000	950	40

Table 2 Paired comparison pilot rankings: formation task

Pilot	Ranking (Best → Worst)	k
1	F2 - NA - F1	1.00
2	F2 - F1 - NA	1.00
3	F2 - NA - F1	1.00
4	—	0.00
5	NA - F2 - F1	1.00

Table 3 Paired comparison pilot rankings: landing task

Pilot	Ranking (Best → Worst)	k
1	L2 - L1 - NA - L3	1.00
2	L1 - L2 - L3 - NA	1.00
3	—	0.20
4	—	0.20
5	NA - L3 - L1 - L2	1.00

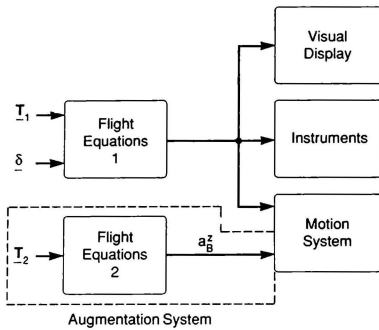


Figure 1 Simulator System

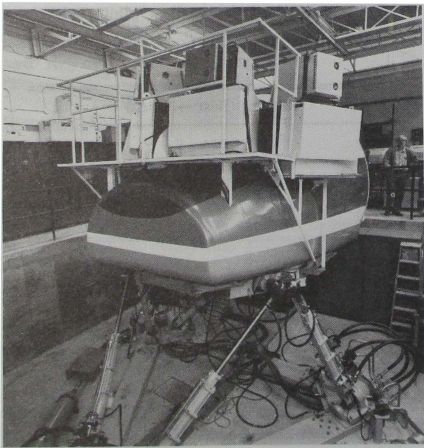


Figure 2 UTIAS Flight Research Simulator

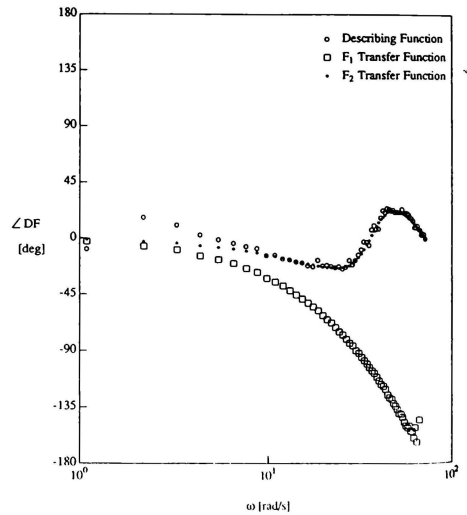
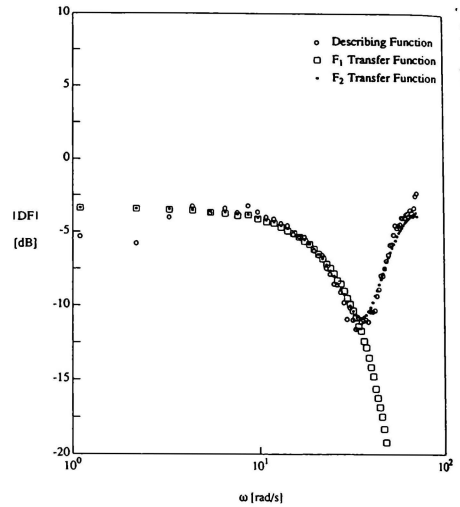


Figure 3 Describing Function for Formation Flight

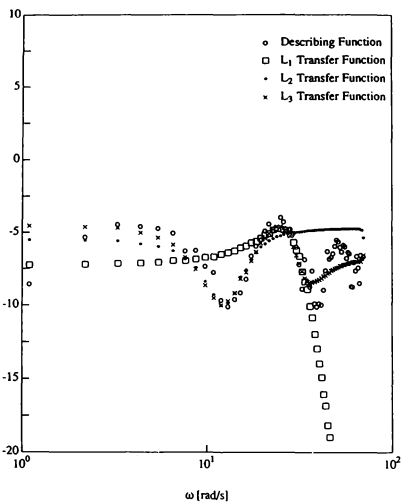


Figure 4 Describing Function for Landing Approach

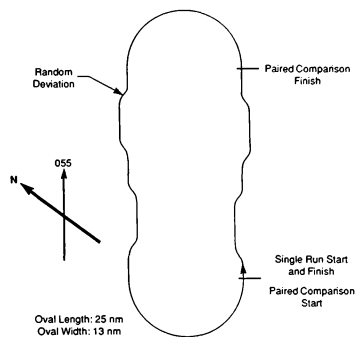


Figure 5 Formation Flying Course

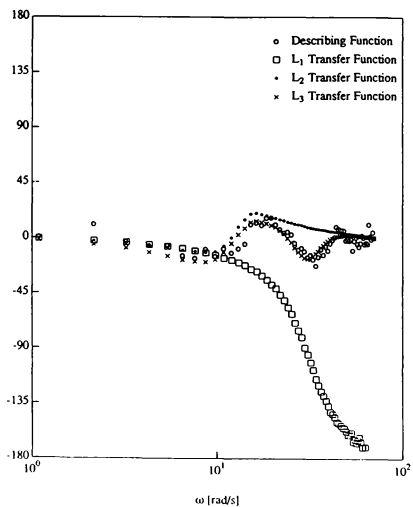


Figure 6 Landing Approach Course

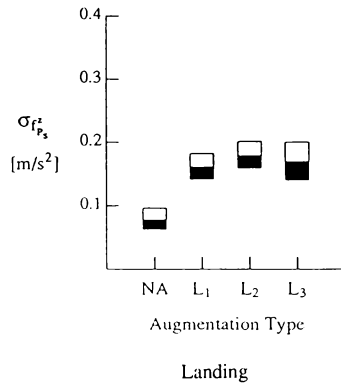
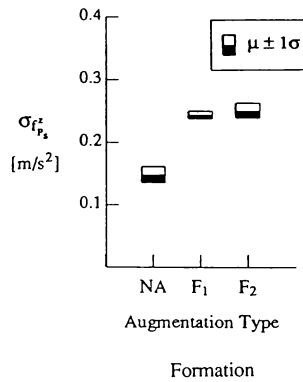


Figure 7 Simulator Heave: Standard Deviation of Specific Force

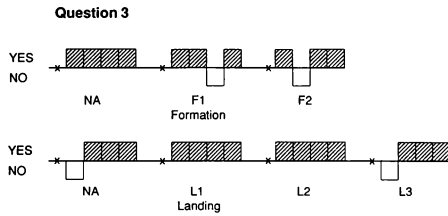
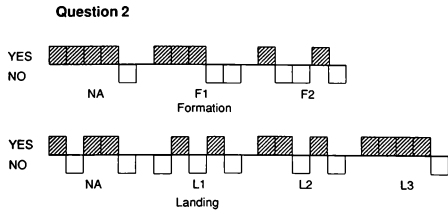
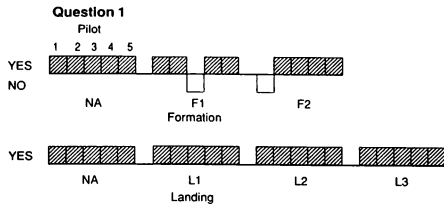


Figure 8 Questionnaire Responses

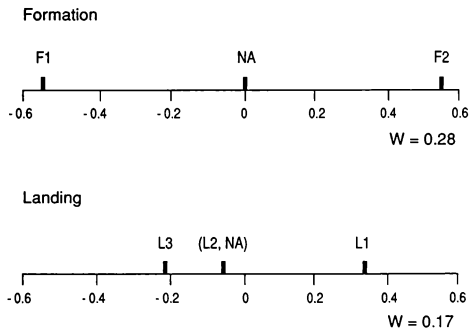


Figure 9 Paired Comparison Ranking

REPRESENTATION OF VEHICLE LOCATION IN NETWORKED SIMULATIONS

Kuo-Chi Lin*, Brian Goldiez**, and Huat Ng†
 University of Central Florida
 Orlando, Florida 32826

Abstract

In a networked simulation environment, it is vital to assure that all vehicles maintain a consistent representation of the geometry between themselves and between vehicles and the terrain over which they operate. Hence, all vehicles should be able to convert their position and orientation to common coordinates. Three coordinate systems, geocentric, geodetic, and topocentric, and the transformation from one system to another are analyzed.

1. Introduction

Distributed Interactive Simulation (DIS) is an exercise involving the interconnection of a number of simulators in which the simulated entities are able to interact within a computer generated environment¹. The simulators may be present in one location or distributed geographically. The communications between simulators are provided by a computer network and the architecture is described in References², which is being developed under an advanced research project sponsored by the Defense Models and Simulation Office (DMSO) and the United States Army Simulation, Training, and Instrumentation Commander (STRICOM).

Traditional approaches to flight simulation have addressed vehicle location internally within the simulation software. Coordination is necessary to assure proper alignment between the simulation software and visual system. With the emergence of networked simulators, the matter of vehicle

positioning becomes more involved. One must now assure that all vehicles maintain a consistent representation of the geometry between themselves and between vehicles and the terrain over which they operate.

There are many approaches to meet the above requirement. The key is to choose a common coordinate system to which every vehicle can transform its own coordinates. Other vehicle positions and relative geometry can then be determined. The choice of the coordinate system should minimize the quantity of the coordinate transformations for computational efficiency. The DIS Standard designates the geocentric coordinates as the common coordinates, or 'world coordinates'. Other world coordinate systems used in the individual simulator include geodetic coordinate and topocentric. The purpose of this paper is to establish the coordinate transformations between these different coordinate systems.

2. Coordinate Systems

There are several different coordinate systems which can be used to describe the position, orientation, and motion of the entities in the simulation. In general, the body-axis coordinates are used to establish the dynamic equations of motion of the entities. Then, an earth-fixed-axis coordinate system is used to describe the kinematics of the entities. This earth-fixed-axis coordinate system is referred to as the world coordinate system and can be considered as an inertial system in our application. All the entities in a distributed simulation environment should use the same world coordinate system. In this paper, only the position coordinates are discussed.

The origin of the geocentric is at the centroid of the earth, with x-axis passing through the Prime Me-

* Assistant Professor, Dept. Mech. & Aero. Eng.,
 AIAA member

** Director, R&D, Institute for Simulation and Training,
 AIAA member

† Associate Engineer, Inst. for Simulation and Training,
 currently with Veda Inc.

meridian at the equator, the y-axis passing through 90 degree East longitude at the Equator, and the z-axis passing through the North Pole. However, in most simulators, a topocentric world coordinate system is used. The earth is basically viewed as flat and the origin is usually placed at the southwest corner of the terrain area with the positive x-axis pointing east, the positive y-axis pointing north, and the positive z-axis pointing up. The definitions of these two coordinate systems can be illustrated in Figure 1.

In order to establish the coordinate transformation between the topocentric and the geocentric coordinates, a third coordinate system, geodetic coordinates, must be introduced. As a matter of fact, some simulators simply use geodetic coordinates as their world coordinates. To define a geodetic coordinate system, the model of the earth needs to be established first.

3. Model of the Earth

If the earth is a sphere, or more precisely, if the equipotential surface mentioned above is a sphere, the spherical coordinates (r, θ, ψ) can be used to describe the location. Figure 2 shows how the spherical coordinates are defined from the geocentric coordinates X, Y , and Z . The transformation between the geocentric and the spherical coordinates can be found in any Calculus textbook³. The geodetic coordinate system for a spherical earth is shown in Figure 3. The transformation between geodetic and spherical coordinates is simply $\phi = \pi/2 - \psi$, $\lambda = \theta$, and $h = r - R_\oplus$, where R_\oplus is the radius of the earth (equipotential sphere). In this case, the transformation between geodetic and geocentric coordinates is also straightforward.

DIS has chosen a more precise model of the earth called the World Geodetic System 1984 (WGS84) in which the surface of the earth is approximated by a reference ellipsoid⁴. This reference ellipsoid is generated by rotating an ellipse about its semiminor axis (axis of the earth), and can be defined by two parameters: the equatorial radius $a = 6,378,137$ meters (the semimajor axis of the ellipse); and the flattening $f = 1/298.257223563$.

If the polar radius (the semiminor axis of the ellipse) is denoted as b , then $b = a(1 - f) = 6,356,752.3142$ meters.

Figure 4 shows the meridional section of the WGS84 ellipsoid, where $W = (X^2 + Y^2)^{1/2}$. The latitude ϕ is defined as the angle between the plane of equator and the normal to the ellipsoid through the point in consideration. In general, the normal to the ellipsoid does not pass through the centroid of the earth.

The Joint Models and Simulation Systems (J-MASS) is an Avionics Simulation Research Program. J-MASS uses a coordinate tree to move between the various reference frames. This program not only says how position information should be shared, but advises users how to move between coordinates. The algorithms, though, are not specified.

4. Coordinate Transformation

One can develop mathematical routines to move between coordinate systems. Simulation imposes some additional requirements. The first requirement is that a level of accuracy must be selected. Second, the computer implementation of the routine must be efficient with respect to time and memory for real time implementation. Third, the number of transformations should be minimized to maintain computation efficiency. Finally, implicit equations must have a consistent solution strategy. The above factors often conflict, making the selection of algorithms difficult.

Assume a Cartesian topocentric coordinate system with its origin defined by the geodetic coordinates (f, l, h) , as shown in Figure 1. The rotation matrix (direction-cosine matrix) from the geocentric to the topocentric coordinates is

$$[R_{dt}] = \begin{bmatrix} -\sin\lambda & \cos\lambda & 0 \\ -\sin\phi\cos\lambda & -\sin\phi\sin\lambda & \cos\phi \\ \cos\phi\cos\lambda & \cos\phi\sin\lambda & \sin\phi \end{bmatrix} \quad (1)$$

The geocentric coordinates (X, Y, Z) and topocentric coordinates (x, y, z) of a point in space must satisfy the following equation:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = [R_{ds}]^T \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} X_o \\ Y_o \\ Z_o \end{bmatrix} \quad (2)$$

where (X_o, Y_o, Z_o) are the geocentric coordinates of the origin of the topocentric coordinate system. Since the origin of the topocentric coordinate system is defined by the geodetic coordinates, we need to establish the transformation from (ϕ, λ, h) to (X, Y, Z) .

Figure 4 shows the meridional section of the WGS84 ellipsoid (the shaded area in Figure 1). Let

$$W = \sqrt{X^2 + Y^2} \quad (3)$$

be a horizontal coordinate in a meridional section. The defining equation of the ellipse in the cross section is

$$\frac{W_e^2}{a^2} + \frac{Z_e^2}{b^2} = 1 \quad (4)$$

Using the geometric relation shown in Figure 4, we can solve for W_e and Z_e as

$$W_e = \frac{a \cos \phi}{\sqrt{1 - (2f - f^2) \sin^2 \phi}} \quad (5)$$

and

$$Z_e = a(1 - f) \sqrt{1 - \frac{W_e^2}{a^2}} \quad (6)$$

Then

$$W_o = W_e + h \cos \phi \quad (7)$$

and

$$Z_o = Z_e + h \sin \phi \quad (8)$$

Finally

$$X_o = W_o \cos \lambda \quad (9)$$

and

$$Y_o = W_o \sin \lambda \quad (10)$$

If the origin of the topocentric coordinate system is defined by the geocentric coordinates, we need to calculate (ϕ, λ, h) from (X_o, Y_o, Z_o) to establish the rotation matrix $[R_{ds}]$.

The longitude angle λ can be obtained directly using

$$\lambda = \tan^{-1} \frac{Y_o}{X_o} \quad (11)$$

The following procedure is used to calculate latitude angle ϕ . The straight line in three-dimensional space passing through (X_o, Y_o, Z_o) and normal to the ellipsoid

$$\frac{X^2}{a^2} + \frac{Y^2}{a^2} + \frac{Z^2}{b^2} = 1 \quad (12)$$

can be defined by the following parametric equations:

$$X = \frac{X_o}{1 + 2m/a^2} \quad (13)$$

$$Y = \frac{Y_o}{1 + 2m/a^2} \quad (14)$$

$$Z = \frac{Z_o}{1 + 2mb^2} \quad (15)$$

where m is the parameter. At the point this straight line intersects the ellipsoid, the parameter m must satisfy

$$\frac{X_o^2}{[a + (2ma)]^2} + \frac{Y_o^2}{[a + (2ma)]^2} + \frac{Z_o^2}{[b + (2mb)]^2} - 1 = 0 \quad (16)$$

Equation (16) can be solved using the Newton-Raphson method. After obtaining the parameter m , Equation (13) ~ (15) can be used to calculate (X_e, Y_e, Z_e) which is the point on the reference ellipsoid corresponding to $(\phi, \lambda, 0)$. Let

$$W_e = \sqrt{X_e^2 + Y_e^2} \quad (17)$$

Then, from Figure 4,

$$\phi = \tan^{-1} \left(\frac{a^2}{bW_e} \sqrt{1 - \frac{W_e^2}{a^2}} \right) \quad (18)$$

and

$$h = \sqrt{(W_o - W_e)^2 + (Z_o - Z_e)^2} \quad (19)$$

5. Universal Transverse Mercator Projection

The topocentric coordinate system treats the earth as a flat surface just like a map does. Since a simulator needs to use the terrain data base in its visual display, it is natural to use a map projection as its (x, y) coordinates. Among all map projection algorithms, Universal Transverse Mercator (UTM) projection is widely used in the simulator industry.

UTM projection is a conformal map projection, and is a modification of the Transverse Mercator

(TM) projection. The earth is divided into grids. Within each grid, (x, y) coordinates are used to specify the points. For the UTM grid system, the ellipsoid earth is divided into 60 longitudinal zones of six degrees each. The zone numbers increase continuously in the eastward direction starting from 180° E. It is intended that a UTM coordinates should be used only in one of the six degree zones plus the overlap area. In each zone, the UTM projection extends for a 40 km overlap into the two adjacent zones. Moreover, UTM coordinates are used only between 80° S and 84° N plus the overlap region. The UTM projection extends to 84° 30' N and 80° 30' S to overlap with the universal polar stereographic (UPS) projections 4.

The easiest way to project a three-dimensional point to a two-dimensional UTM projection point is by using geodetic coordinates. The process can be performed in three steps 4:

- (1) The position of the point (ϕ, λ, h) is projected onto the reference ellipsoid $(\phi, \lambda, 0)$.
- (2) Find the UTM zone using the geodetic longitude λ .
- (3) Convert coordinates (ϕ, λ) to (x, y) .

The formulas for the conversions between (ϕ, λ) and (x, y) can be found in References 4,5.

There is no simple and accurate algorithm for transforming between geocentric and UTM coordinates available yet.

6. Conclusions

From the above derivations, it is evident that the key for all the conversions is the geodetic coordinate system. The utility of geocentric coordinates is not clear from a mathematical or practical point of view. IST, though, has developed specific computer routines for real time implementation of the preceding mathematical routines for moving from geocentric to geodetic to topocentric coordinates.

References

1. "Military Standard (Final Draft) Protocol Data Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation." Technical Report IST-PD-91-1. Institute for Simulation and Training, University of Central Florida, Orlando, Florida, May 1992.
2. "Draft, Communication Architecture for Distributed Interactive Simulation." Technical Report IST-CR-92-6. Institute for Simulation and Training, University of Central Florida, Orlando, Florida, May 1992.
3. Anton, H., "Calculus", John Wiley & Sons, New York, NY, 1988, 3rd ed., pp. 862-863.
4. DMA, "Datums, Projections, Grids and Common Coordinate Systems, Transformation of." Military Handbook MIL-HDBK-600008. DoD., May 1991, pp. 38-49.
5. Snyder, J., "Map Projections - A Working Manual" U. S. Geological Survey Professional Paper 1395, 1987, pp. 48 -65.

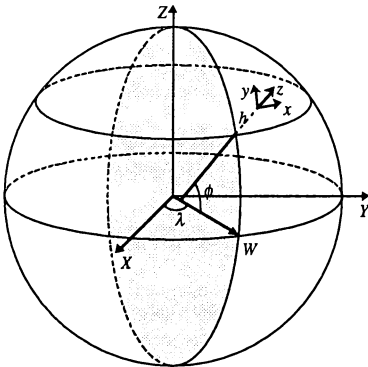


Figure 1. Geocentric and topocentric coordinate systems.

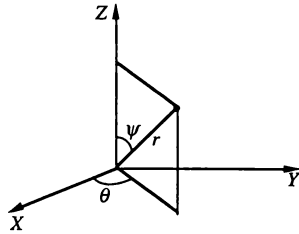


Figure 2. Spherical coordinate system.

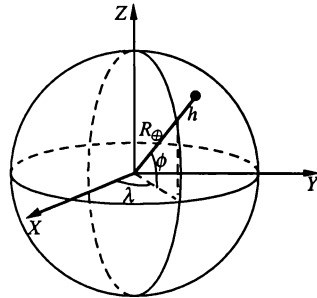


Figure 3. Geodetic coordinate system (spherical earth).

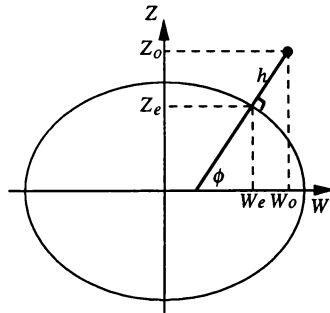


Figure 4. Meridional section of the WGS84 ellipsoid.

IMPLEMENTATION OF EXPERT SYSTEMS WITHIN AN INTERACTIVE TACTICAL ENVIRONMENT

D. N. Siksik and R. D. Rourke
CAE Electronics Ltd.
St-Laurent, Quebec, Canada

ABSTRACT

The requirement for an interactive and realistic environment is present in simulation for both training and research. A necessary factor in achieving this environment is the ability to model the intelligence of the entities within it, thus providing a capability for realistic behaviour and interaction.

CAE's Interactive Tactical Environment Management System (ITEMS) provides simulations with an environment in which entities are closely modelled with respect to both systems and intelligence, allowing them to interact with each other and with their environment in a realistic manner. ITEMS approaches the problem of simulating entity intelligence through the implementation of expert systems.

The further improvement of networking geographically remote simulations together and to ITEMS provides several enhancements including the capability for more cost-effective and realistic training.

INTRODUCTION

A piloted flight simulator almost always flies in some kind of a computer synthesized environment. For commercial passenger jet simulators, this environment consists mostly of the terrain and features of the visual system's database and possibly weather. The synthesized environment is the *reality* in which the pilot finds him or her self flying in.

The elements comprising the environment of a commercial flight simulator are not required to react with the simulated aircraft. For example, the simulated traffic that a pilot may see during a night flight would not change if the simulated aircraft passed a few meters above them. The behaviour of the cars (or lights) is hard coded to move in a set pattern. A

simulation environment in which the behaviour of the elements is not affected by what the simulated airplane does is classified as *non-interactive*.

Most first generation tactical simulators could be considered as having a non-interactive environment. Typically, these simulators could provide training missions in which the crew would attack some threat elements that were pre-programmed to follow a set route or remain stationary. The behaviour of the threats would not depend on the actions of the simulated vehicle. Many model board simulators are of this type.

The next generation of flight simulators include threats that can react to the piloted vehicle. For example, the US Army's first Apache simulators incorporated enemy air defence vehicles that could search and track the simulated Apache and shoot it down. It is not possible, however, for the Apache to affect the threats, which must be killed or deactivated by a training instructor. Another drawback of this level of simulation is that it provides an unrealistic many versus one training situation. Neither full interaction nor team training or team tactics are possible. Since the interaction in this environment is unidirectional (threat to piloted vehicle), it may be classified as *semi-interactive*.

CAE Electronics Ltd. has taken several approaches towards creating a *fully-interactive* tactical environment for simulation. Such an environment should provide interaction between all elements and their environment and would enable realistic training or effective research in both one versus one as well as team tactics.

These approaches have included:

- 1) The addition of more piloted vehicles to the simulated environment. To minimize costs, workstations equipped with joysticks could

be used in lieu of high fidelity flight simulators. The extra vehicles could play the role of friend or foe to the flight simulator.

- 2) The addition of fully interactive computer generated players to the environment. CAE's Interactive Tactical Environment Management System (ITEMS)^[1,2] provides simulations with an environment in which entities are closely modelled with respect to both systems and intelligence allowing them to interact with each other and with their environment (terrain/atmosphere/ocean) in a realistic manner.
- 3) The networking of geographically remote simulation platforms with each other and with ITEMS. The resulting environment includes computer generated players, human-controlled vehicles as well as remote players (which may be piloted flight simulators or computer generated players in their own right). A *seamless* level of simulation requires all players, whatever their origins, to appear and interact in an authentic manner.

THE CREATION OF AN INTERACTIVE TACTICAL ENVIRONMENT

The tactical environment provided by ITEMS is both controlled and created by the user. Through the use of a database management system (DBMS), scenario files, each of which represents a complete tactical environment, may be individually created, modified or downloaded to the host-simulation computer to be run at real-time.

The scenario design function of ITEMS is an off-line process involving the ITEMS Database Management System (DBMS)^[2]. During scenario design, the user provides the DBMS with the information required to simulate the tactical scenario. In order to hold large amounts of data efficiently, the DBMS is divided into individual libraries. Information about scenarios, players, systems, intelligence, etc. is stored in respective libraries as individual records. The libraries are organized in a hierarchical format so that high level libraries can reference the lower level ones. For example, the specifications defined for a gun round in the Gun Rounds Library may be referenced

by a gun in the Gun Library which may be referenced by an aircraft within the Player Library. Finally, the aircraft may be referenced by a scenario within the Scenario Library. Any scenario in this library may be executed.

Players represent the most basic elements of a tactical scenario and are defined as any entity which has tactical importance. A player could represent elements such as tanks, trucks, installations, SAM sites, infantry, fixed wing, rotary wing, stealth craft, ships, submersibles, etc.

The ITEMS scenario combines players within a terrain and visual database in which they may interact realistically with each other as well as with such environmental elements as weather or ground features (see figure 1).

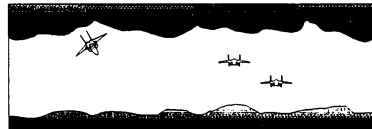


Figure 1. An out-of-cockpit view of a tactical simulator as part of a formation being attacked within a tactical scenario.

An example may consider a Tornado tactical flight simulator flying lead to a formation of computer generated Tornado aircraft. The formation is sensitive to wind and rain conditions as well as ground features and will react to the approach or attack of an enemy MiG-29 aircraft (see figure 2).

In order to achieve realistic player interaction, ITEMS implements representative modelling of:

- Dynamics
- Navigation
- Ballistics
- Systems
- Intelligence

This modelling is based upon detailed knowledge

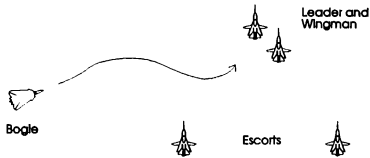


Figure 2. Formation of Tornados with Leader represented by a tactical simulator and Wingman and Escorts by computer generated players.

of physical player data. For its definition, a player references numerous low level library records within DBMS. These represent its platform and various systems. Platform definition is made with respect to physical characteristics, dynamic envelope and vulnerability. Systems libraries provide specification data for modelling and include: active and passive sensors (radar, FLIR, RWR, LWR, etc.), weapons (guns, rockets, missiles, bombs, etc.), countermeasures (flares, smoke, jammers, etc.), communications (voice, data, visual, etc.) and laser (designator, range finder). By combining platform and systems references, a large variety of computer generated players are created. For example, the computer generated wingman and escorts in figure 2 would be provided with definitions for crew-eyes. These definitions would specify identification ranges and bearings (visual cone) as well as degradation factors based upon a decrease in visibility (rain, fog, etc.).

The creation of lifelike player behaviour and reaction, however, requires the modelling of player intelligence and this, in turn, is based upon knowledge of player tactics (military doctrine). Tactics, whether used in air, ground or naval applications, require a specialized range of expertise. ITEMS exploits the ability of expert systems to carry out tasks of an expert nature thereby providing tactical

capability to players.

Within the ITEMS expert systems, each type of tactical knowledge is represented by IF/THEN rules and is called a "doctrine". This knowledge provides control over the actions of individual players as well as over the summary actions of groups, such as a change of formation. Doctrines, like player data, are organized into libraries within the DBMS and are referenced by players within the scenario. ITEMS provides the following doctrines:

- (1) Mission Doctrine:
Knowledge pertaining to the mission of a player (goals, routes, contingencies, etc.).
- (2) Prime Opponent Selection Doctrine:
Criteria for the selection of a prime opponent for the player in question.
- (3) Air Combat Doctrine:
Pilot level knowledge controlling the selection of manoeuvres and weapons during air combat.
- (4) Command and Control Doctrines:
Doctrines applied to players organized into command structures such as battalions and companies. These doctrines include coordination and control functions such as the cooperation of players in a company towards a common goal and the control/change of company formations based on situation.

The specification of the above doctrines as rules is based upon the generation of appropriate parameters relating to the field of interest. These parameters constitute the condition and response part of the rule. The ITEMS expert systems use a combination of research and interviews with subject-matter experts in order to define parameter representations of specific domains. The parameters chosen are typically high-level and representative of compiled knowledge, thus allowing the user to create rules without the intervention of a knowledge engineer.

Examples of typical high level parameters for air combat include:

Negative_Aspect_Angle_and_Increasing.
Positive_Aspect_Angle_and_Decreasing.
Leader_About_to_be_Tracked

Parameters take into account the specialized

nature of the doctrine (tactics) with which they are associated so that wingman related parameters are referenced with respect to the formation leader.

Such parameters would be used in creating an air combat doctrine to be assigned to the wingman in figure 2. A possibility might be for the wingman, once it has detected a hostile aircraft, to scan for the possibility of its leader being tracked. The doctrine assigned to the wingman could then issue a voice communication (via standard radio, as defined for that player) to the leader. Since, in this case, the leader is a manned tactical simulator, the pilot would hear a digital voice message saying "*Break Left*", for example, and would be able to react to the threat which his computer generated ally detected.

It is possible to extend the concept of team training or interaction to the level where two pilots are work as wingman and leader. The wingman in the above example could be replaced by a geographically remote flight simulator. Wingman and Leader could thus interact over long-haul network. The ability to extend a tactical environment to networking is discussed in the next section.

DISTRIBUTED INTERACTIVE SIMULATION

As previously mentioned, CAE's latest approach to creating an interactive tactical environment that best meets training and experimental needs involves the connecting of high-fidelity simulators together into a network. This is possible because of the previous research sponsored by DARPA in developing a distributed simulator called Simulation Network or SIMNET. Results of this work led to the development of the Distributed Interactive Simulation (DIS) standard version 1.0^[3,4]. DIS is a public domain standard that defines what information a connected simulator must send and expect to receive and at what rate. This section will discuss the problems and benefits of making a simulator DIS compatible.

The term for the software and the hardware that interfaces a simulator to a DIS network is "gateway". Our gateways uses the DIS standard to "packetise" local information about the tactical environment and send it on the network. The packets in DIS are called Protocol Data Units (PDU), and they contain information about the state of players and weapons.

The definition of the DIS PDU set is strongly motivated by the desire to minimise the load on a wide-area packet switch network that stretches thousands of kilometres. Figure 3 graphically illustrates the size of the projected DIS network in the United States. One result of this motivation was the decision to use a system of "dead-reckoning" for a positional data.

A dead-reckoning system uses old information, and assumed information to estimate where something is. In terms of DIS, the old information is the last known location and attitude of an element, and the assumed information is the element's speed, and accelerations. By integrating speed and accelerations, a DIS dead-reckoning system estimates the position of a scenario element.

A DIS gateway uses dead-reckoning to decide when to update information on a local element. In other words, it compares the actual position and attitude of an element to what the other sites think it is. If the errors in attitude or position exceed a well defined limit, it sends a new PDU. The DIS standard also requires that a gateway sends a PDU on an element no less than a pre-set time limit of five seconds so that other sites will not assume that it is dead.

A problem encountered during trials was that the ITEMS ground vehicles where generating to many PDUs. A ground vehicle will pitch and roll while moving. The average amplitude of these attitudes is a function of speed, vehicle size and ground roughness. As the vehicles sped up during an engagement they would flood the network with PDUs. The solution to this problem involved filtering the ground vehicles attitude so that they only reflected the ground slope.

The solution to the problem of the ground vehicles generating many PDUs is only a temporary one. If an environment has many airplanes flying in it, and they are experiencing turbulence then the same problem will probably occur. Also, by removing the effects of air density and ground roughness, the simulation fidelity is decreased. A better solution that CAE proposes is to send the filtered attitudes and add a new field to the PDUs that give information on the degree of roughness that could be used by visual systems to recover the effects.

Dead-reckoning is also used by a gateway to process received data. Obviously the image of a

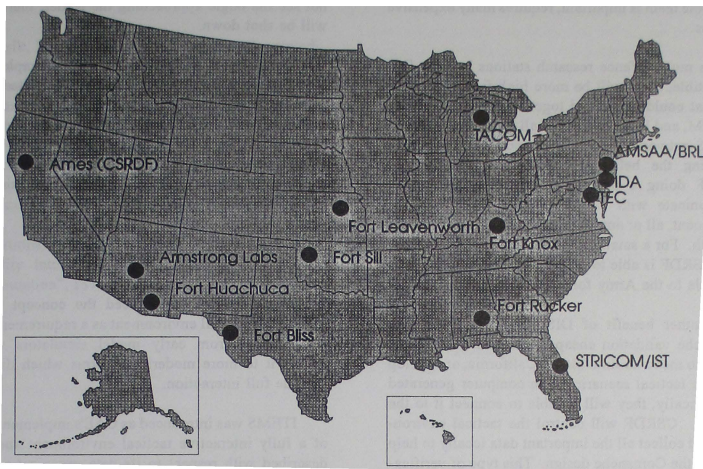


Figure 3. Current or potential BDS-D nodes

threat would look very jumpy on a high-fidelity visual systems if its position was updated at less than 60 Hz. In such a case it would be easy for the pilot to determine which elements in the environment were external and the system would not be seamless.

We have been doing much work on overcoming the problem of jumpy targets in the visual. Special algorithms that are not part of the DIS standard were developed. The problems we have seen with the received PDUs include packet ordering, incorrect data, and of course time delays. Our smoothing algorithm overcome most of these problems.

The Crew Station Research and Development Facility at NASA Ames is a US Army laboratory for conducting research into human-machine interfaces of crews to rotor-crafts and tactical environments. To conduct this type of research, CSRDF requires a high fidelity simulator (i.e. with a blade element rotor model and an IG) that is connected to the type of environment provided by ITEMS with a DIS gateway. Work done at CSRDF plays an important part in the design and development of the Comanche helicopter.

The ITEMS system at CSRDF was the first to have a DIS gateway. A prototype of this system was demonstrated as part of the I/ITSEC conference in

San Antonio Texas in November 1992. An important part of this conference was the integration of many simulation manufacturers systems onto a common local DIS network that was used to demonstrate a large air-land-sea battle. CAE's prototype provided an Apache for that demonstration. This demo was the first large scale verification of DIS with different manufacturers using the same DIS network at the same time.

In May 1993, at the AUSA conference at Orlando Florida, the finished CSRDF site in California, and sites in Fort Rucker Alabama and Stratford Connecticut where connect via long haul network to ITEMS simulators in Orlando. This DIS set-up ran a large air-land battle simulation.

Most experiments performed at a research establishment like CSRDF involves computer generated forces, experimenter operated players and fully crewed simulators. The high-fidelity re-configurable generic crew station at CSRDF is an excellent vehicle for giving an experiment the effect of a human in the decision making loop. The relationship between human errors and work-load is very important, and can only be tested in a high-fidelity station as opposed to a low-fidelity single workstation player. To perform experiments into "team" tactics, in which

the whole team is important, requires many expensive stations.

As more defence research stations become DIS compatible, there will be more high-fidelity simulators that could be pooled together. When CSRDF, TACOM, and Fort Rucker are all fully functional on the DIS network, an experiment could be developed involving the helicopters from Fort Rucker and CSRDF doing a passage of lines in which they communicate with TACOM tanks. For such an experiment, all or any of the sites could be doing the research. For a small investment into a DIS connection, CSRDF is able to greatly improve the facilities available to the Army for research.

Another benefit of DIS is that it will make Comanche validation cheaper. Instead of Sikorsky having to move its simulator to California, or develop complex tactical scenarios with computer generated forces locally, they will be able to connect it to the network. CSRDF will control the tactical environment and collect all the important data locally to help validate the Comanche design. This type of verification can be call "Remote Verification."

DIS in Training

DIS can also improve interactive tactical environments used in training as well as experiments. The US Army plans to use DIS technology in their upcoming Close Combat Tactical Trainer CCTT project. CCTT will involve connecting many tank trainers together on a network with computer generated forces. The main use of DIS for training will be in team training. To illustrate this idea we will present an example of a training scenario using DIS.

Figure 3 shows the lay-down of players in the example scenario. It consists of four bombers (Tornados) in escort formation. The lead bomber is the local piloted simulator and its wingman is a DIS player. The two other bombers are computer controlled.

To teach the wingman the importance of keeping an eye on the rest of the formation, the instructor increases the separation between the two sets by slowing down the computer controlled bombers. Once they have fallen far enough back that they would be unable to see a bogie approaching the leader, the instructor reposition an IAT threat near the leader and lets it attack. If the leaders wingman does

not see the bogie approaching the leader, the leader will be shot down.

The purpose of the training, in this example, is to teach the two forward bombers the importance of team work. For this type of training to be useful, both the leader and wingman must be real crews in high-fidelity simulators that provides an accurate level of work load and information. Networking two Tornado simulator with an ITEMS environment would facilitate this level of team training.

OVERVIEW

This paper has introduced the concept of an interactive tactical environment as a requirement that has evolved from early model simulators which lacked it to more modern platforms which did not provide full interaction.

ITEMS was introduced as CAE's implementation of a fully interactive tactical environment and was described with respect to its data organization and modelling capabilities in the goal for realistic representation. Emphasis was placed upon the use of expert systems to model player intelligence, this approach providing for the life-like behaviour of computer generated players.

The extension of an interactive tactical environment to the networking of geographically remote simulation platforms was introduced. Problems in implementation were discussed as well as the substantial benefits of providing cost-effective team training and more realistic levels of interaction.

ITEMS is currently implemented on both training and research simulation facilities around the world. The expert system capabilities are applied to all manner of computer generated forces, including air targets and providing capabilities such as intelligent wingman or adversary. Applications also include system and research support functions such as rule driven data recording and analysis.

REFERENCES

1. Morris A., "The Creation of Complex Tactical Training Environments, an Unclassified Approach", 10th Interservice/Industry Training Systems Conference, pp. 151-160, November

1988.

2. Siksik, D. N., "Intelligent Computer Generated Forces Through Expert Systems", Third Conference on Computer Generated Forces and Behavioral Representation, Orlando, Florida, March, 1993.
3. "Protocol Data Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation", Version 1.0, IST-PD-91-1.
4. "Rational Document; Entity Information and Entity Interaction in a Distributed Interactive Simulation", IST-PD-90-1.

AIAA-93-3585-CP

**PSEUDO AIRCRAFT SYSTEMS:
A MULTI-AIRCRAFT SIMULATION SYSTEM FOR AIR TRAFFIC CONTROL RESEARCH**

Reid A. Weske
SYRE

George L. Danek
ATC Field Systems Office

NASA Ames Research Center, Moffett Field, CA

Abstract

Pseudo Aircraft Systems (PAS) is a computerized flight dynamics and piloting system designed to provide a high fidelity multi-aircraft real-time simulation environment to support Air Traffic Control research. PAS is composed of three major software components that run on a network of computer workstations. Functionality is distributed among these components to allow the system to execute fast enough to support real-time operation. PAS workstations are linked by an Ethernet Local Area Network, and standard UNIX socket protocol is used for data transfer. Each component of PAS is controlled and operated using a custom designed Graphical User Interface. Each of these is composed of multiple windows, and many of the windows and sub-windows are used in several of the components. Aircraft models and piloting logic are sophisticated and realistic and provide complex maneuvering and navigational capabilities. PAS will continually be enhanced with new features and improved capabilities to support ongoing and future Air Traffic Control system development.

Introduction

Pseudo Aircraft Systems is a stand-alone software system that simulates a multi-aircraft environment for use in the test and evaluation of automated air traffic management and control systems. The objectives of such systems are to enhance safety, improve traffic flow, reduce fuel consumption and decrease the workload of air traffic controllers. They do so by providing computer-generated advisories to assist controllers in managing the traffic in their areas of responsibility. An air traffic controller's primary source of information is a large radar display that shows a plan view of the geographical region under his or her control. Superimposed on the display are "blips" representing the locations of aircraft flying in the region; attached to each blip is a text block that displays information such as airspeed and altitude. The air traffic controller uses this information to assess the traffic situation and, based on judgement, experience and any available automated aids, issues instructions to the pilots to manage traffic flow and spacing.

The purpose of PAS is to simulate traffic scenarios so that automated Air Traffic Control systems can be tested and evaluated in real time and under realistic conditions. PAS simulates aircraft that respond to controllers' instructions and provides corresponding aircraft state data to generate the blips and other information presented on their radar displays. The geographical region included in a typical simulation contains an Air Route Traffic Control Center (ARTCC) and a Terminal Radar Approach Control (TRACON) Facility. The ARTCC and TRACON control different airspace areas as illustrated in Figure 1.

The first version of PAS was developed in the 1970s at NASA Ames Research Center as an adjunct to air traffic control research in progress at the time. The original software was written in FORTRAN and ran on a XEROX Sigma 9 mainframe computer. Eventually, this computer became obsolete, so in 1986 the software was reprogrammed in C and migrated to a network of Sun Microsystems workstations. System enhancements were also implemented at this time, including the capability to simulate a larger volume of air traffic and a greatly expanded airspace. Another major upgrade was initiated in April, 1992 to provide additional capabilities required to support more complex and sophisticated testing. This revision provided the capability to handle multiple airports and a theoretically unlimited number of aircraft. Additional enhancements included more efficient and flexible user interfaces for the operators, simplified setup procedures, and improved operational control over the simulated aircraft suite.

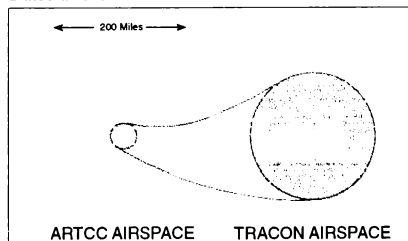


Figure 1. Air Traffic Control Areas

This paper presents an overview of the current version of PAS from the user's perspective. The system is described in terms of its three major components, details about the functions of each component and important implementation issues. Operational deployment and usage are then discussed, and finally, future development possibilities are touched upon.

PAS System Description

System Overview

The major components of PAS are three distinct sets of programs that run on a network of computer workstations. These components are called the Simulation Manager, the Pilot Station and the Pilot Manager; a pictorial representation of the network is shown in Figure 2. Each of the components is operated and controlled by means of its own custom designed Graphical User Interface (GUI). The Simulation Manager is the heart of the system and runs on a single workstation. It serves as the control center for the overall aircraft simulation and as the computer link between the PAS network and the Air Traffic Control system. The Simulation Manager also contains the kinematic models of the individual aircraft comprising the simulated traffic scenario. The kinematic models process commands to produce aircraft state variables which are used to generate the simulated radar track data, the primary outputs of the system. The Pilot Station software runs on a set of workstations that serve as both input devices and information display devices

for operators called pseudopilots. Each pseudopilot controls a number of individual aircraft by issuing commands in response to verbal instructions from air traffic controllers or electronically transmitted datalink commands, much as an actual pilot controls his aircraft. The commands are pre-defined strings of alphanumeric characters that are entered using a standard workstation keyboard. The Pilot Manager serves as an interface between each Pilot Station and the Simulation Manager and also as an administrator for the set of Pilot Stations. It relays commands from all the Pilot Stations to the Simulation Manager and conversely, aircraft state data from the Simulation Manager to the appropriate Pilot Station.

Because of its primary function as a research support tool, PAS was designed to be flexible and easy to use to allow rapid prototyping of changing test conditions. New airspace regions, wind conditions, aircraft types or traffic scenarios can be implemented by altering data only; the algorithms remain unchanged. As currently structured, PAS software imposes no limit on the number of aircraft or the number of airports that can be simulated.

Simulation Manager

The functions of the Simulation Manager are simulation initialization and control, dynamic modelling of the aircraft comprising the scenario, and managing the flow of information between PAS and the client Air Traffic Control system for which it is furnishing simulated aircraft. The Simulation Manager also performs system level error checking and notification. The Simulation Manager is comprised of two separate and distinct processes, and has two functional modes called Initialize and Operate.

The Simulation Manager Initialize mode provides the means to set up the overall operational environment for the simulation and to start and stop real-time execution. The environment is configured and initialized by specifying (1) a database that models the airspace to be simulated, (2) a file that lists initial conditions for the set of aircraft that will initially be included in the simulation, such as position, speed and the time at which each will become active, (3) wind conditions and (4) an optional set of run history files. When the simulation is started, the mode changes from Initialize to Operate, the PAS clock is activated and real-time simulation commences.

In Operate mode, the software performs functions to monitor and control the simulation and to alter the scenario in real time. Specific control functions are provided to (1) interrupt the flow of aircraft into the scenario at any time (stop new aircraft from becoming active), (2) delete the next aircraft scheduled to be-

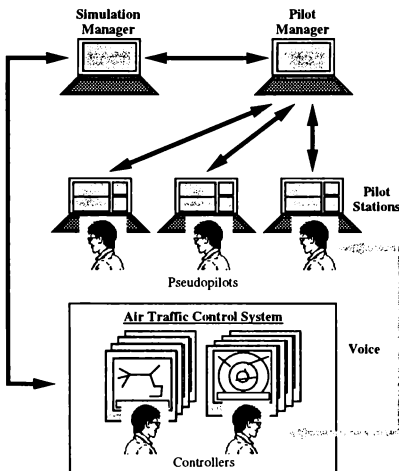


Figure 2. PAS Elements and Connections

come active without affecting the remainder of the set, (3) allow additional aircraft to be created and inserted into the scenario while the simulation is running and (4) change the runway designation for inactive aircraft.

The Simulation Manager monitor function is provided by a Flight Data Table that displays information about active and inactive aircraft using electronic flight progress strips similar to the strips used by air traffic controllers. The Flight Data Table also provides a tool to remove active aircraft from the simulation.

Aircraft kinematic models and piloting logic are combined in the PAS Aerodynamics process. Kinematic models representing both commercial transports and private aircraft are available. Each aircraft in the scenario is modelled separately. Aircraft accelerations are computed in three degrees of freedom based on external forces, and these are integrated twice and transformed to the appropriate coordinate system to represent radar track data.

The piloting logic part of PAS Aerodynamics is a complex control algorithm that guides each aircraft along the desired flight path in response to commands from the pseudopilots. The piloting logic processes pseudopilot commands to generate appropriate inputs to the aircraft model, monitors the resulting aircraft motions and generates further control adjustments to null any errors between the desired and actual flight path.

Pilot Station

The Pilot Station software provides the features and functions necessary for a pseudopilot operator to issue flight commands to a number of aircraft and to monitor their flight status. A pseudopilot can control only a limited number of aircraft at any given time (10 or 12 is a reasonable maximum for typical scenarios), so most simulations require several workstations, each running the Pilot Station software for a different pseudopilot. The workload is distributed by allocating additional workstations to *sectors*, which are specific geographical regions in the vicinity of the airport. The Pilot Station software contains three separate processes, and the same two functional modes as the Simulation Manager—Initialize and Operate.

The Pilot Station Initialize mode is similar to the Simulation Manager Initialize mode except that it provides the means to set up the piloting environment for a specific pseudopilot workstation rather than the environment for the entire simulation. This environment is configured and initialized by specifying (1) the same airspace database that was selected in initializing the Simulation Manager, (2) the Pilot Manager-

to-Pilot Station communications link, (3) the route designated for the aircraft to follow to navigate to the airport and the associated altitude and speed conditions that must be maintained and (4) the sector name within which the workstation can control aircraft (its area of authority).

In Operate mode, the pseudopilot workstation serves as the operator's station for the simulated aircraft which are assigned to it. It provides the means to "fly" the aircraft and monitor their status; that is, the functions necessary to perform the simulated flying task. Numerous commands are available to the pseudopilot including a series of vectoring commands (commands that change heading, altitude or speed), scheduling commands (commands that reassign a flight path or establish the aircraft on a non-radar route) and certain special commands (holding patterns, missed approaches, runway and airport changes and transfers from one controller sector to another).

PAS also provides the capability to simulate datalink commands, the automated commands transmitted electronically from the Air Traffic Control computer system to the aircraft. Any datalink command received by the Pilot Station is displayed; the pseudopilot has the option of accepting or rejecting it. There is also an option to accept (that is, implement automatically) all incoming datalink commands without pseudopilot action.

An Instrument Panel display presents essential aircraft information required by the pseudopilot to monitor the status and progress of the aircraft under his or her control. This display presents, for each aircraft, the aircraft call sign, a code identifying the type of aircraft, the actual and commanded magnetic heading, the actual and commanded altitude, the actual and commanded indicated airspeed, the actual and commanded Mach Number, the Estimated Time of Arrival or other aircraft route information and the airport and runway assigned for landing.

A Command History display shows the commands (text strings) entered by the pseudopilot, exactly as typed, and commands that have been validated and implemented (these latter commands are echoed back from the Pilot Manager).

Overall situational awareness of the airspace in which the aircraft are operating is provided by a Radar Display, which is a map-like plan view of the local airspace. This display portrays the position of the aircraft in the simulation scenario as well as certain ground features and other information using symbology that is similar to that of an Air Traffic Controller's radar display. A number of features are available to allow the pseudopilot to customize the presentation charac-

teristics of the Radar Display, including panning, zooming in and out, selecting alternate sectors and several declutter modes.

The Radar Display also includes a PAS error/warning text area that displays notifications that may originate in any of the PAS components. These include operating system error messages, PAS error messages and warnings to indicate that an invalid command was entered.

Pilot Manager

The Pilot Manager provides executive control over the set of Pilot Stations and acts as a transfer station between the components of the PAS network. It links the Simulation Manager and the Pilot Stations and passes aircraft state data in one direction and aircraft commands in the other. The Pilot Manager consists of three separate processes, and has the same two Initialize and Operate modes.

The Pilot Manager Initialize mode configures and initializes the overall piloting environment. This is accomplished by specifying (1) the same airspace database that was selected in initializing the Simulation Manager, (2) the Simulation Manager-to-Pilot Manager communications link, and (3) setup conditions for a powerful support function called the Auto Controller. The Pilot Manager also coordinates the initialization of Pilot Station workstations.

In Operate mode, the Pilot Manager serves as an executive Pilot Station. In addition to the capabilities of a normal Pilot Station, it can also command any of the active aircraft, rather than only those assigned to a specific sector. The Pilot Manager also performs administrative functions such as managing the transfer of control responsibility for individual aircraft from Pilot Station to Pilot Station (called a "handoff" in Air Traffic Control jargon).

The Auto Controller is a special purpose function designed for use in TRACON simulations only. It initializes and controls a specified number of aircraft flying in the ARTCC airspace so that they approach and enter TRACON airspace in a realistic manner. This is all accomplished without human intervention, thereby eliminating the need for pseudopilots acting outside the TRACON airspace. The aircraft are automatically issued descent and arrival commands that mimic the arrival instructions issued by an Air Route Traffic Control Center controller; for example, a Standard Terminal Arrival Route. These commands are stored in the file selected during initialization of the Auto Controller. The final command issued by the Auto Controller to each aircraft is a handoff to one of the TRACON Pilot Stations.

Implementation Issues

Networking and Communications

At NASA Ames, PAS runs on a homogenous network of Sun Microsystems workstations connected via an Ethernet Local Area Network. The minimum configuration of the system is illustrated in Figure 3, where each square represents a separate workstation and the circles represent software processes. This minimum configuration requires three workstations. A more typical operational configuration, however, requires two to four Pilot Station workstations or four to six workstations total, since each additional Pilot Station requires a separate workstation. As stated earlier, each Pilot Station controls a specific sector in the simulated airspace. If further reductions in pseudopilot workload are necessary, up to four Pilot Stations can be assigned to each sector. When used in this manner, the program automatically distributes the aircraft in a sector uniformly among the number of assigned Pilot Stations, so that each Pilot Station workstation controls only certain aircraft within the assigned sector. At present, PAS software can accommodate a maximum of ten sectors or a total of forty Pilot Stations; the system is easily configured to handle additional Pilot Stations in the Pilot Manager initialization process.

Aside from the requirement to service multiple pseudopilots, the main reason for implementing PAS on a network of workstations is because a single workstation cannot support real-time operation. The maxi-

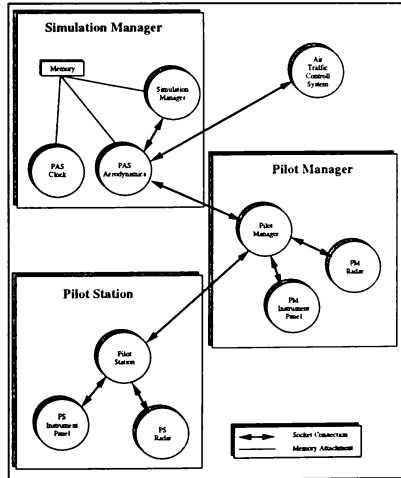


Figure 3. PAS Network Schematic

mum throughput of the CPU is insufficient to execute the entire system in real time, and in addition, the Sun operating system does not provide real-time control features. By allocating the PAS processes to three different workstations there is less competition for CPU resources, and sufficient time is available to complete all the required computations.

PAS data transfer and inter-computer communications are implemented using standard UNIX socket communications protocol. All sockets are of the stream socket type (not to be confused with AT&T UNIX streams). This method provides sequenced, reliable, full-duplex connection-based byte streams. For machine-to-machine communication, the ARPA Internet protocol is used, and for communications among processes on the same workstation, the UNIX internal protocol is used. One advantage of using the UNIX internal protocol is that port numbers need not be assigned, thereby reducing the likelihood of port collisions with other applications on the network.

PAS uses only two primary data transmission transactions: (1) *aircraft state data* are sent from the Simulation Manager to the Pilot Manager, the Pilot Stations and the client Air Traffic Control System and (2) *aircraft command data* are sent from a Pilot Station through the Pilot Manager to the Simulation Manager. The data block size for each aircraft is 256 bytes. Aircraft state updates originate in the PAS Aerodynamics process and are therefore controlled by the Simulation Manager. The cycle time for this process is set at one-half-second to assure valid aircraft state data. Data are transmitted to the other PAS components and to the Air Traffic Control system at a much slower rate (once every four seconds) as this is sufficient and similar to the update rate of an actual Air Traffic Control computer. This latter transmission frequency is called the system update rate and is configurable in increments of whole seconds.

Unlike aircraft state data, command data can be received by the Simulation Manager at any time, as they are transmitted whenever a command is issued from a Pilot Station. Command data is sent in two parts; a transaction header of 84 bytes followed by an optional variable length packed data buffer.

Graphical User Interfaces

The original impetus for using Graphical User Interfaces (GUIs) for PAS was that relatively inexperienced and untrained users would be called upon both to start up and monitor the system and to serve as pseudopilots. GUIs were the only practical method to provide an operating environment that was both easy to use and capable of controlling and monitoring such a complex system. Reference 1 provides complete

descriptions of all the PAS GUIs and includes detailed instructions on their use.

The Sun Microsystems XView tool box and the X Windows System were chosen to program the current PAS GUIs. X Windows is the most popular windowing system on UNIX platforms. PAS developers were experienced in both X Windows and XView, and the earlier version of PAS used SunView. Therefore, implementing the GUIs in XView promised to be a relatively straightforward task. In addition, X Windows has the capability to display its graphics on a remote terminal.

Although a separate GUI was developed for each component of PAS, common features and design techniques were used wherever possible to simplify the development process and produce a more consistent and easier to use product. *Widgets* are used in the same manner regardless of where they appear. *Buttons* are used as a toggle, to invoke a function, or to display another window. *Menus* are used to present a list of reasonable options to the user. It should be noted that menus simplify the input validation process significantly. *Text fields* are used to input information from the keyboard when a menu is impractical to build. *Drawables* are used when appropriate pre-defined widgets are not available.

Each GUI is composed of a set of windows, some of which are similar or even identical for the three software components. All three GUIs contain similar Initialization Windows which provide the means to implement the initialization and setup functions described earlier. Figure 4 shows the Initialization Window for the Simulation Manager, which is the most complicated of the three, as an example. The Pilot Station and Pilot Manager Initialization Windows also serve to spawn and monitor the processes running on their respective workstations, while the Simulation Manager GUI uses a unique Simulation Control Window to implement these functions. This window also serves to implement some of the real-time control features. The Simulation Manager Operate mode GUI

Figure 4. Simulation Manager Initialization Window

windows are all unique to this component. These are (1) the Flight Data Table described earlier, (2) the Halt Traffic Window, (3) the Change Runways Window and (4) the Create Aircraft Window.

Conversely, the Operate mode GUI windows are identical for the Pilot Manager and Pilot Station except for one functional difference—aircraft commands issued from the Pilot Manager are not restricted to a certain sector, but can apply to any aircraft in the simulation. The windows (illustrated in Figure 5) that implement the Operate mode functions described earlier are (1) the Pilot Messages Window, (2) the Radar Display Window and (3) the Instrument Panel Window.

Aircraft Models and Piloting Logic

PAS uses a three degree of freedom point-mass model to compute the translational dynamic characteristics of the simulated aircraft. In addition, a roll degree of freedom is included (but not externally controlled) to provide more realistic behavior in turns. The model employs airspeed, flight path angle and heading angle

as state variables; mass is the only inertial parameter included. Aircraft body axis accelerations are calculated using realistic lift, drag and powerplant forces, weight and wind effects. The accelerations are integrated twice to produce corresponding velocities and displacements which are transformed into the ground track (North and East) and altitude parameters that represent radar track data. Additional details pertaining to dynamic modelling in PAS, including complete development of the equations of motion, are provided in Reference 2.

PAS currently contains aerodynamic and powerplant data for six commercial transport aircraft, a representative turboprop aircraft and a representative single engine private aircraft. The aerodynamic data include the effects of flaps, spoilers, landing gear and compressibility. Powerplant data are provided by a special engine modelling routine and are derived from actual aircraft engine data.

PAS also provides a sophisticated array of piloting and control functions that allow the aircraft to perform a wide variety of maneuvers with minimum pseudopilot

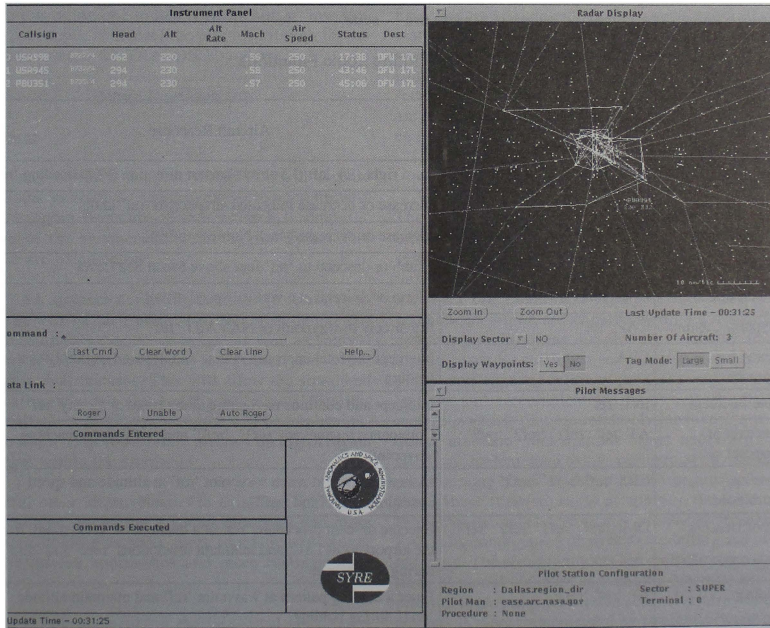


Figure 5. Pilot Station Operate Mode GUI

workload. The piloting logic is implemented as a closed loop control system with the aircraft model as the controlled component. This implementation results in smooth, realistic maneuvers because changes in state are effected by integrating the equations of motion. Reference values for the control parameters are derived from the pseudopilot command, and a control algorithm executes the appropriate actions to control the response of the aircraft model to achieve the maneuver commanded by the pseudopilot. The piloting logic can perform both simple maneuvers, such as a changes in altitude, or very complex maneuvers, such as holding patterns or automatically navigating an airway. PAS provides twenty-nine such piloting functions and many of these have several variations.

Pseudopilot commands are in the form of strings of alphanumeric characters called *command strings*. Command strings are entered either by typing in the characters using the standard workstation keyboard and keypad, or by depressing one of the function keys which are programmed to enter the most commonly used command strings. The command strings are designed to be as short as possible and mnemonic where possible. A typical command string consists of one or two fields containing the command designator and a reference value. The command string is always pre-

ceded by a field designating the specific aircraft to be affected (the aircraft call sign). For example, "UAL333 R270" is the string that would be entered to command the aircraft with call sign UAL333 to execute a right turn to a heading of 270°. Some of the more common and interesting piloting functions and their corresponding command syntaxes are listed in Table 1.

PAS Usage

The primary use for PAS at the present time is to support the development of an Air Traffic Control System known as the Center-TRACON Automation System (CTAS), which is under evaluation by NASA and the Federal Aviation Administration (FAA) as a replacement for current automated air traffic scheduling tools. A brief overview of CTAS tools is provided in Reference 3, and a more complete description of the system's design aspects is provided in Reference 4. PAS is used to simulate the aircraft under the control of CTAS in test and evaluation sessions. Most of this work is conducted at NASA Ames Research Center located at Moffett Field, California. PAS is also used to support related CTAS development efforts at the FAA Technical Center in Atlantic City, New Jersey and MIT Lincoln Laboratory in Lexington, Massachusetts.

Table 1. PAS Piloting Logic Functions

Command		Aircraft Response
Type	Syntax	
Heading	R'ref', L'ref', F'ref'	Turn right (R), left (L) or by shortest direction (F) to heading 'ref'
Speed	S'ref'	Increase or decrease indicated airspeed to 'ref' knots
Mach Number	M'ref'	Increase or decrease Mach Number to 'ref'
Altitude	A'ref'	Climb or descend to 'ref' feet above Mean Sea Level
Vertical Speed	DSR 'ref'	Climb or descend at a vertical speed of 'ref'
Waypoint Capture	CAP 'ref'	Fly directly to waypoint or NAVAID 'ref'
Cleared for Approach	CLA	Intercept the final approach course and descend smoothly to a landing
Route Intercept	INT 'ref'	Intercept and continue navigating along route or airway 'ref'
Maneuver at Waypoint	AT 'ref' 'ref1' 'ref2' 'ref3'	Commence maneuvers 'ref1', 'ref2' and 'ref3' upon reaching waypoint 'ref'
Crossing Restrictions	CRS 'ref' 'ref1' 'ref2'	Maneuver so as to reach waypoint 'ref' at altitude and speed conditions 'ref1' and 'ref2'
Top of Descent	DME 'ref' 'ref1' 'ref2' 'ref3'	Execute descent procedure 'ref' at a distance of 'ref1' n. mi. from the airport VORTAC, and maintain restrictions 'ref2' and 'ref3' during the descent
Holding	HL 'ref' 'ref1'	Enter a holding pattern at waypoint 'ref' and maintain altitude 'ref1' during holding
Missed Approach	MA	Execute a missed approach

All current PAS operations fall into one of two categories: TRACON simulations and Center simulations. A TRACON simulation covers only the TRACON airspace (see Figure 1). It typically includes two "feeder pseudopilots" who control the aircraft from the TRACON-ARTCC airspace boundary to the final approach area, and two more "final pseudopilots" who control the aircraft from the final approach area to touchdown. In addition to these Pilot Station workstations, PAS also requires two workstations for the Simulation Manager and Pilot Manager components. The corresponding CTAS configuration for the simulation consists of an air traffic controller position for each pseudopilot, and two additional workstations to implement CTAS scheduling and sequencing tools and system management functions. The total number of workstations required for the simulation, therefore, is twelve.

An ARTCC simulation covers the ARTCC airspace and requires essentially the same computer environment. There are typically one to two "high sector" pseudopilots controlling aircraft from the ARTCC airspace boundary to an intermediate point at which the aircraft begin the descent from cruise altitude. From this point, one to two corresponding "low sector" pseudopilots control the aircraft through the arrival descent to the TRACON boundary.

Future Developments

Features

Like any research support tool, PAS will require continual upgrade and improvement to keep up with the changing demands of the research it supports. Some of the enhancements anticipated in the near future are listed below.

The FAA airspace definition database (referred to as ACES) will be incorporated to establish actual airways (both "J" and "V" airways) as well as navigational aids, intersections, waypoints and other geographical features. This will allow the simulated aircraft to navigate along real world routes. Similarly, realistic departure procedures will be implemented to allow PAS to simulate aircraft taking off from airports within the airspace.

Several other improvements to existing capabilities are planned, including a wider variety of and more realistic holding patterns, more sophisticated and realistic missed approaches and more realistic wind effects.

Finally, enhancements to the Graphical User Interfaces and incorporation of object-oriented design methodologies are planned.

Applications

Enhancements to PAS will allow researchers to develop departure scheduling procedures that can be used with arrival scheduling tools to provide for full system (that is, from takeoff to landing) aircraft scheduling capabilities. Scheduling appropriate *departure* times so aircraft fit into the eventual arrival flow and minimize (or eliminate) en route delays would lead to significant savings in both time and energy.

Another advanced application would be a full Air Traffic Control system arrival scenario that would allow simultaneous evaluation of both ARTCC and TRACON scheduling tools. Aircraft would enter the simulation at the boundary between two ARTCCs and be controlled through the ARTCC (high and low sectors) to the TRACON boundary. The aircraft would then be handed off to TRACON controllers and pseudopilots and progress through the feeder and final sectors to touchdown.

PAS could also serve as an effective training tool for use in Federal Aviation Administration facilities across the country (ARTCCs, TRACON Facilities and the FAA Training Academy). Current pseudo-aircraft simulation tools are simply not realistic enough to support training in the use of state-of-the-art automation aids. Limitations on the number of aircraft and controller sectors and a reliance upon actual controller displays are several of the shortcomings that would be eliminated by using PAS.

Summary and Conclusions

The PAS multi-aircraft simulation system is a valuable tool for supporting Air Traffic Control research. The system provides a wide range of functions and features, and the GUIs make it easy to learn, set up and use. PAS functionality is separated into three major components which are implemented on a network of computer workstations. Employing a network in this manner provides sufficient computational resources to permit real-time execution of the software. The workstations and communications hardware are reliable, mature products that are relatively inexpensive to acquire, maintain, and upgrade when necessary. The GUIs are implemented using commercial off-the-shelf programming tools, so the same design attributes apply to these software items as apply to the hardware. Probably the most technically advanced feature of PAS is its combination of aircraft modelling and piloting logic. The aircraft models are realistic representations of actual airliners and smaller aircraft, and the piloting logic provides a comprehensive array of maneuvers that are implemented by straightforward control actions. PAS has been used for more than a decade to support the development of Air Traffic Control

systems at Ames Research Center. The use of PAS has expanded over the past several years to other facilities that have become involved with this and related research. A number of upgrades and improvements are planned to support ongoing test and evaluation efforts. Additional enhancements will be implemented to keep pace with future demands.

References

1. Weske, Reid, "Pseudo Aircraft Systems (PAS) Users Manual", Release 3.1, SYRE, NASA Ames Research Center, Moffett Field, CA, May 1993.
2. Mukai, Conrad, "Design and Analysis of Aircraft Dynamics Models for the ATC Simulation at NASA Ames Research Center", Seagull Technology, Inc., Sunnyvale, CA, March 1992.
3. Erzberger, Heinz, CTAS: Computer Intelligence for Air Traffic Control in the Terminal Area, NASA Technical Memorandum 103959, July 1992.
4. Erzberger, Heinz, Davis, Thomas J. and Green, Steven "Design of Center-Tracon Automation System", presented at the AGARD Guidance and Control Symposium on Machine Intelligence in Air Traffic Management, Berlin, Germany, 11-14 May 1993.

A RADAR ALTITUDE AND LINE OF SIGHT ATTACHMENT

Russell Sansom
Senior Graphics Programmer
SYRE, a division of SYSCON Services, Inc.
NASA-Ames Research Center, Moffett Field, California

David Darling
Senior Graphics Programmer
SYRE, a division of SYSCON Services, Inc.
NASA-Ames Research Center, Moffett Field, California

Abstract

This paper describes a method of overcoming much of the computational expense of finding radar altitude and determining lines of sight in flight simulations over databases built from polygons. Methods are described for quantizing polygonal databases and for searching through them quickly. Various tuning parameters are explained and run-time performance figures are offered.

I. Introduction

Problem

Finding terrain altitude and determining lines of sight over polygonal databases of the kind used in digital image generators can be computationally expensive. This is because the general solution for finding terrain height requires a comprehensive search. Every polygon in the database must be tested to determine if it lies above the X-Y position where terrain height is to be calculated. Determining if the line of sight (LOS) is clear between two points requires a similar search.

When multiple terrain altitudes or LOS determinations are required for a simulation mission, the burden on an image generator can become excessive, leading to a compromise in the number of LOS calculations delivered or to changes in the allocation of resources within the image generator itself; not usually a practical solution.

Solution

A successful body of software has been created at NASA Ames Research Center's Vertical Motion Simulator facility for generating quantized data bases and traversing them quickly on a computer which is auxiliary to the image generator. The simulation mainframe can receive radar altitude, LOS, and range information from the Radar Altitude and Line of Sight Attachment whether the image generator is enabled or not.

Considerable energy has gone into creating a system that will work with the data from various image generators and can be tuned to a broad range of research requirements. The software design was also driven by several other general

principles: It had to be easily modified, comprehensible by other professional staff members, and free from obscure programming tricks.

Overview

This paper describes a method for pre-calculating a grid of terrain altitudes and for using this grid to determine radar altitudes and lines of sight at minimal computational expense.

It first discusses means of acquiring polygonal data from a database development system. It then explains a method for turning these polygon vertex data into a Digital Terrain Elevation Data (DTED) database, sometimes referred to as a "bed of nails."

Next, the simple calculation of terrain altitude and the technology for determining LOS and range are explained. Finally, some advanced applications for the Radar Altitude and Line of Sight Attachment are described.

II. Methodology

Data Acquisition

A Digital Terrain Elevation Data (DTED) database is created by finding the terrain altitude of every point across some pre-defined grid of points. A typical Defense Mapping Agency map, for example, uses satellite radar altitudes, aircraft photos, and even first person observations to define every altitude across a grid spacing of 3 arcseconds latitude by 3 arcseconds longitude on the globe's surface.

The principle behind data acquisition for the Radar Altitude and Line of Sight Attachment is similar. A grid is defined under an arbitrary grouping of polygons. The vertical distance between the ground plane and the overlying polygon at each grid intersection is calculated. As each of these elevations is found, it is stored in a two-dimensional array. Figure 1 shows a database in its original polygonal form and as a "bed of nails" after quantization.

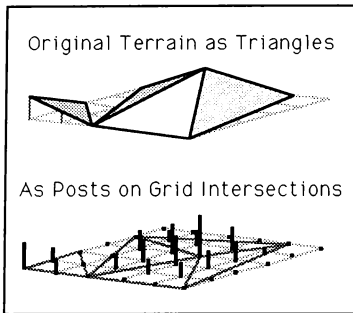


Figure 1 Two Database Representations

A polygonal database is transformed into a DTED database by a compiler which accepts a textual description of polygons as input and generates a file containing a two-dimensional array of elevations as output. This data acquisition phase has been broken into two main modules, the parser or "front end," and the DTED module.

Parser Module

Digital image generators build up a scene from lists of polygons supplied in textual form by a Database Development System. This textual list is normally used as the source code for an image generator's database compiler.

The parser module needs to perform two tasks:

- 1) find the points which define each polygon and
- 2) dissect those polygons into triangles

Finding Points

In a typical example from the Evans and Sutherland CTSA system, a line of source code resembles a DEC PDP Macro-11 statement:

```
.poly POLY1 point1, point2, point3, point4
```

This source code statement says there is a polygon* named POLY1 which is comprised of the four vertices point1, point2, point3, and point4. Because this is a single pass compiler, the definitions of points must already have been created in statements resembling these:

```
.pnt point1 100 , 100 , 20
.pnt point2 200 , 100 , 20
```

* In the E&S CTSA system a polygon is a convex figure located in three-space consisting of between 3 and 8 coplanar vertices.

```
.pnt point3 250 , 200 , 45
.pnt point4 150 , 200 , 45
```

The parser processes these *.pnt* directive lines by the straightforward method of searching each line of input for the string *.pnt*. When this string is encountered, the name is stored in a character table. The *.pnt* directive is checked for the correct number of arguments. If all points are not properly defined the compiler prints an error message and halts. The X, Y, Z coordinate strings following the string *.pnt* are converted from their ASCII representations to their numerical values and stored in a table indexed to their name.

When a *.poly* directive is encountered, each of its arguments is located in the name table and the corresponding X, Y, Z coordinates are placed in a temporary polygon storage structure along with the vertex count in readiness for dissection into triangles.

Dissection into Triangles

Theoretically, the DTED module could process polygons with any number of vertices. However, in order to completely decouple the parser from the DTED module, the atomic triangle form was chosen. The reduction into triangles also avoids the problems that scan line algorithms have with filling concave polygons.

When the parser module is called upon by the DTED module to supply a triangle, the parser supplies a triangle from its current temporary polygon storage structure. If this structure is exhausted, the parser will resume searching the source file for another polygon. Polygons are reduced to triangles (figure 2) with a simple algorithm:

```
for N = 3 to Num_Verts
  Next_Triangle = vert(1), vert(N-1), vert(N)
```

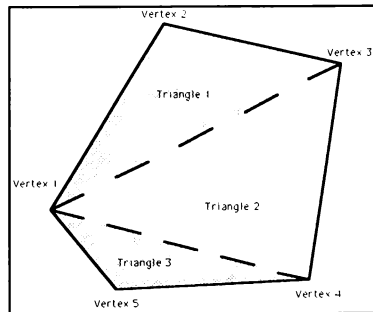


Figure 2 Reduction of a Polygon into Triangles

This example represents the parser's major functions. Front ends have been written for Singer-Link DIG1 raw data dumps, "documentation" output from the MultiGen* modeling system, and Evans and Sutherland CTSA source code.

A few complications have been encountered in writing various parsers. In cases where the image generator's description language allowed macros, instancing, and similar indirect approaches to polygon creation, it has been necessary to write parser code that expands the macros or generates individual polygons from instancing directives. It has also been feasible to modify the actual Image Generator's compiler to expand the macros into intermediate code which then was used as input to the parser. A much "smarter" parser module is also required when coordinate system transformations are imbedded within the database description.

The parser's task is simply to supply one triangle after another to the routine which calls it. The DTED module has no interest in what polygon or what grouping of polygons a given triangle comes from. These liberating concepts allow both modules to be completely decoupled from each other. When revisions are made to one module, they can be carried out with the confidence that they will not affect other modules in the package.

DTED Module

The actions of the DTED module and the parser are interwoven. Once a triangle is obtained from the parser, the DTED module will process it completely into the final world-coordinate output array before calling on the parser for another one. When the DTED module has processed all available triangles, this final array is written to disk for later use by the run-time module.

Preliminary Steps

Before the DTED module can execute, the programmer must choose the size of quantization steps. An elevation will be calculated every "stepsize" distance in the X and the Y directions. There are several tradeoffs which determine the best step size for a particular research mission.

1. The smaller the step size, the more memory used.
2. The smaller the step size, the smaller the quantization error.
3. The smaller the step size, the slower the LOS algorithm runs.
4. Step size has no effect on time to determine radar altitude.

Figure 3 offers a summary of memory requirements for various database sizes where altitudes are stored as 16-bit integers.

The raster-scan algorithm of the DTED module requires a private X-Y triangle buffer which has been dimensioned to contain the largest triangle that will be encountered in the input data. These dimensions can be satisfied with an intelligent overestimate or by a slightly revised version of the parser/DTED module which inspects every triangle and prints out the maximum X and Y dimensions encountered. Three other dimensions which have to be known ahead of time are the DTED array sizes (X and Y), and the .pnt name hash table size.

Step Size in meters	Database size in km	Memory Required in bytes
100 m	1 x 1	200
100 m	4 x 4	3,200
100 m	20 x 20	80,000
10 m	1 x 1	20,000
10 m	4 x 4	320,000
10 m	20 x 20	8,000,000
1 m	1 x 1	2,000,000
1 m	4 x 4	32,000,000
1 m	20 x 20	800,000,000

Figure 3 Memory Requirements vs. Database Size

When the data acquisition program first starts running, the entire DTED array is usually initialized to zero to represent a default ground plane. If an existing database is to be combined with a new one, then the DTED array is initialized to the existing database by reading it in from a disk file.

Processing Loop

The processing loop of the DTED module must now find altitudes beneath each triangle. One of the principle problems in this process is finding which of all the X-Y coordinates in the DTED array lie inside or acceptably near the boundaries of the triangle. While it would be feasible to check every DTED coordinate against every triangle in the database with an "inside" test, on a large database such a procedure could require many hours to complete. Furthermore, the inside test could not identify coordinates which lay outside the triangle boundaries, but were within one-half a grid space of it. The processes described in this section were designed to identify all the correct X-Y coordinates near each triangle and to circumvent the computational expense of repeated comprehensive searches.

Moving a Triangle into the Triangle Buffer

Before a triangle is moved into the triangle buffer, the buffer is initialized to some unique value which cannot be

* The MultiGen modeling system is a product of Software Systems, San Jose, California

confused with terrain altitude later by the raster-scan algorithm. Any negative number is an appropriate "Not_An_Alt" value.

To simplify buffer management, the triangle is biased so its leftmost vertex is moved to the first column of the triangle buffer its lowest vertex is moved to the first row.

Once these minor details are arranged, a literal image of the triangle is drawn into the triangle buffer. Each intersection in the buffer which is crossed by one of the triangle's boundary lines is set to the actual altitude of the line using the method explained below. All the remaining intersections in the buffer are left at their default, "Not_An_Alt" value (figure 4).

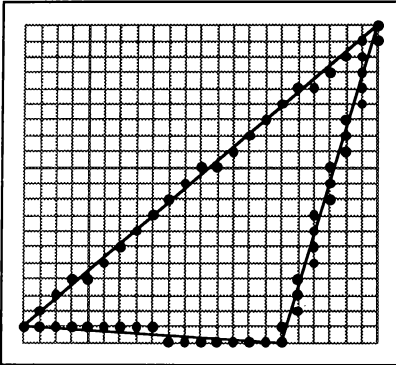


Figure 4 Triangle Image Written into Triangle Buffer

DDA Algorithm Used to Choose Steps Along Line

The method for deciding which intersections in the triangle buffer are "touched" by the triangle's outline is a very important technical detail. According to Newman and Sproull¹, a computer-generated line should:

- 1) appear straight,
- 2) terminate accurately,
- 3) have constant density,
- 4) be drawn rapidly, and
- 5) have a density independent of line length and angle.

The choice of the Digital Differential Analyzer (DDA) from the graphics discipline seems obvious because it fulfills all these requirements for drawing proper lines. Newman and Sproull point out that the most difficult of these requirements is to create constant density no matter what the angle or the extent of a line because a line-length estimate is required. Because of the raster-scan algorithm used in a later step there may be no "holes" in the line. A correctly-

formed estimate of line length will draw either exactly the correct number of dots or one more dot than necessary, but never fewer.

There are many variations of DDA to choose from. A three-dimensional extension of a two-dimensional symmetrical DDA² was created for this task. Once tuned to its purpose, the DDA is the essentially correct approach to digitizing lines. A pseudocode explication of the DDA mechanism is given here:

DDA($X_1, Y_1, Z_1, X_2, Y_2, Z_2$)

(NOTE: This algorithm assumes that both points are in the first quadrant ($X \geq 0, Y \geq 0$))

[Check if points are out of bounds]

(Total change in X,Y,Z)

$dX = X_2 - X_1$; $dY = Y_2 - Y_1$; $dZ = Z_2 - Z_1$

(number of iterations)

Steps = floor(max(dX, dY)) + 1

$dX_Step = dX / Steps$ (dX per step)

$dY_Step = dY / Steps$ (dY per step)

$dZ_Step = dZ / Steps$ (dZ per step)

(Accumulators for positions)

$X_Accum = X_1 + 0.5$ (Offset by 0.5...)

$Y_Accum = Y_1 + 0.5$ (for rounding)

$Z_Accum = Z_1 + 0.5$

indexed loop from 0 to Steps

(Truncate Accumulators)

$Pixel_X = \text{floor}(X_Accum)$

$Pixel_Y = \text{floor}(Y_Accum)$

$Pixel_Z = \text{floor}(Z_Accum)$

(Take action for this pixel - Set its altitude, check LOS blockage, set its color, etc.)

(Increment Accumulators for next Step)

$X_Accum = X_Accum + dX_Step$

$Y_Accum = Y_Accum + dY_Step$

$Z_Accum = Z_Accum + dZ_Step$

end loop

end DDA

Plane Equation Coefficients are Found

Altitudes found in the subsequent steps will be calculated using the plane equation:

$$ax + by + cz + d = 0. \quad (1)$$

Assuming an input triangle described by the three vertices

$$(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3)$$

the coefficients (a,b,c,d) of the plane containing this triangle are:

$$\begin{aligned} a &= (x_2 - x_1) * (z_3 - z_4) - (z_2 - z_1) * (y_3 - y_1) \\ b &= (y_2 - y_1) * (x_3 - x_1) - (x_2 - x_1) * (z_3 - z_1) \\ c &= (x_2 - x_1) * (y_3 - y_1) - (y_2 - y_1) * (x_3 - x_1) \\ d &= -a(x_1) - b(y_1) - c(z_1) \end{aligned}$$

Buffer Lines Scanned to Find Coordinates Inside Triangle

Once a proper outline of the triangle has been placed in the triangle buffer, a very simple raster-scan scheme is used to determine all the buffer addresses which fall inside the triangle boundaries. The buffer is scanned from left to right, one row at a time, looking for the first and the last valid altitude on a line. Having found the left and the right extremes of the scan line, a loop is set up between ROW_{leftextreme} and ROW_{rightextreme} to calculate the elevation at each column. Elevations are calculated using the plane equation (1). The following information is known:

- (a,b,c,d) coefficients of the plane found earlier.
- (x,y) the current step in the raster-scan algorithm.

Therefore:

$$cz = -ax - by - d$$

$$z = \frac{-ax - by - d}{c}$$

The Z thus found is placed into the triangle buffer at the current X,Y, replacing the token "Not_An_Alt."

The Processed Triangle is Written to DTED Array

Once a triangle has been completely processed into the triangle buffer, it is biased back to its proper world-coordinate position and at each location it is compared to the value already in the DTED array. If the new value is higher, it replaces the old value. Since the "Not_An_Alt" token in the triangle buffer is a negative number, only positive altitudes will be transferred to the DTED output array.

Final Steps

When the parser module cannot fetch more data from the input file, processing is complete. The DTED array is written out to disk for use by the run-time module.

Features of Resultant Database

While the X-Y dimensions of the DTED array are fixed, the number of polygons processed may be any number from none to millions. Besides its obvious use as a radar altitude hash table, the resultant data base has some other useful and important properties.

Databases are Easy to Combine

DTED databases used for finding radar altitudes are built primarily from terrain. However, if the grid spacing is fine enough to represent small details adequately, an array of trees, cultural features, or fixed military threats can be created separately with the data acquisition software. Subsequently this array can be combined with a terrain array. This is accomplished by a program which reads the two arrays to be combined, compares each X,Y location to find the taller feature, and creates a resultant array with the winner of the comparison in each location.

This separation into a terrain data set and a tree data set has been useful when, for instance, the terrain is satisfactory but trees have had to be re-located several times to meet specific mission requirements.

It is also feasible to adjust the DTED array dynamically with run-time code to include moving features such as threats.

DTED Arrays Make Useful Displays

DTED arrays can be turned into high-quality static displays with a graphics program that plots every point in the array as a color proportional to the altitude at that point. Since the dynamics of CRT's are easily overwhelmed by the large range of altitudes in a typical map, the graphics programmer must exercise some cleverness in choosing hues and intensities for a given representation. This method is, however, quite straightforward and immediately rewarding. If a hard copy printer is available, the resultant bit map can be visually inspected for correctness and unexpected aberrations.

To verify a real world database (such as one generated from DMA satellite data) the hard copy image can be compared to an actual topographical contour map.

Two independent DTED arrays that are meant to be similar can be compared by generating a hard copy of each and overlaying them on a light box. In some cases this may be one of the few methods available to verify that several bit maps that are supposed to be functionally identical are indeed the same.

Run-Time

The run-time module is decoupled from the software which generates its database; this is an important element of its design. The software will work identically over data generated by the data acquisition software, actual DMA data arranged into an X-Y array and rationalized to a flat earth, or

a hand-generated "bed of nails." Nothing is required except an X-Y array of data.

When the Radar Altitude and Line of Sight attachment was first implemented several years ago, it was designed to operate as a standalone program on a dedicated DEC MicroVAX so that no memory or cycle time penalty would be imposed on the main simulation computer. The MicroVAX was sent X-Y-Z positions for the own-ship and target by the mainframe simulation computer over an Ethernet connection. The MicroVAX returned variables containing the aircraft's radar altitude, the target's radar altitude, the X-Y-Z position of the point which blocked the Line of Sight, a flag signifying the inter-visibility of the two points, and information about possible errors. The simulation mainframe calculated the hypotenuse between the occulting point and the first or second ship position to obtain range information.

Radar Altitude

Calculating the radar altitude of a point (typically the CG of the simulated aircraft) is extremely simple.

Where:

HCG is the aircraft altitude above sea level

and

Terrain_Height at point (X,Y) is simply the contents of the DTED array at address (X, Y).

then

$RADALT = HCG - Terrain_Height.$

Execution Speed is Independent of Database Size

Except for the penalty imposed by some computer architectures on looking up large arrays, the time to fetch one terrain height is totally independent of database size and resolution. For this reason it is computationally economical to build a DTED database with the finest resolution memory will allow.

Interpolation May be Necessary for Some Missions

On some research missions the DTED array post spacing may be too coarse for smooth instrument readout. Slow helicopter flight over coarsely spaced altitude posts is a good example. At a ground speed of 20 knots over posts spaced every 100 meters, the radar altimeter would jump to a new discrete reading every 0.1 seconds.

Discrete stepping of terrain height in the run-time module is alleviated by interpolating among the four nearest posts. Interpolation is first performed from north to south on the east side of the aircraft. A second interpolation is performed from north to south on the west side. Finally, an interpolation is performed from east to west between these two points. This creates a saddle on non-coplanar posts as opposed to a hill or a valley which can result from three-point interpolation (figure 5).

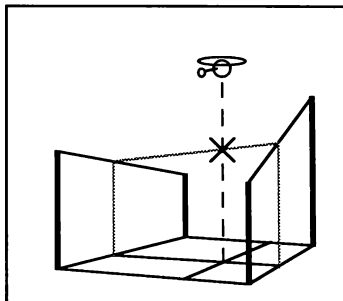


Figure 5 Bilinear Interpolation of Radar Altitude

Line of Sight

The LOS portion of the run-time software must determine if any terrain blocks the view of the target from the own-ship. It does this by performing an exhaustive search of all the DTED posts standing between the two points. For various reasons it may also be of interest to know how far the occulting object is from one eyepoint or the other.

The Line of Sight algorithm generates a three-space line between the own-ship and the target. The terrain altitude stored at each X-Y location passed over by the line is fetched and compared to the line's altitude (its Z). Terrain height is found by looking it up in the DTED array. The line's altitude is generated by successive steps of a three-dimensional DDA. If the terrain is lower than the line, the search continues. If the terrain is higher than the line, the search terminates and the current X, Y, Z position is returned for use as range data. As an option, when the first search terminates another search can be instigated from the second eyepoint to determine the range to its occulting terrain.

DDA Technology is Used for Traversing the Posts

The DDA is the perfect tool for defining the three-space lines required by the LOS module. It automatically generates the next nearest DTED intersection to interrogate without concern for duplication, omission, or having to perform a mathematical rounding function every time it is called. As in the DTED module, the line length estimate is made equal to or more than the number of intersections so that no intersections will be missed. It does not require repetitive multiplication and it generates the correct Z along the line without the continual need for a slope-intercept formula.

Parameters Need Tuning

The LOS search is most computationally expensive when the line between the two points is not interrupted by terrain so that the entire search must be conducted. In order to hold search times on large data bases to a practical value the allowable distance between the own-ship and the target eyepoints is limited to some maximum which is agreed upon by the researcher. If the two points violate this maximum in a preliminary check, then the LOS is declared automatically blocked and an error condition is flagged in the return data packet.

Average execution time for the LOS algorithm can be reduced by lowering the greatest allowable search distance. In general, execution time is linearly proportional to the maximum number of posts traversed by a specific line in either the X or the Y direction. That is:

Where Loop_Time is the time to interrogate one post,

$$\text{Total_Time} \sim \text{Loop_Time} * \max(|Nbr_Posts_x|, |Nbr_Posts_y|).$$

This leads to the somewhat surprising fact that a given line length will take the longest time to search if it lies along the X or the Y axis. The least search time will result from a line which lies at a 45-degree diagonal. Note that the grid intersections along the 45-degree diagonal to the X-Y axis are spaced 1.414 times greater apart than the dots on a line laying along one of the axes. Therefore, the speed gained in searching a diagonal line is compensated for by lowered resolution. Figure 6 shows the execution times for repeated LOS calculations on some well-known computers.

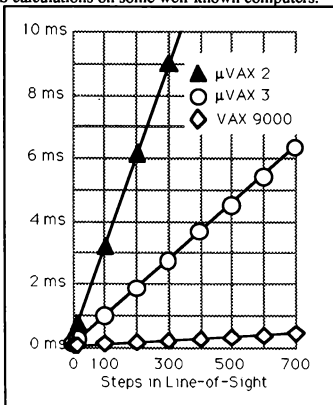


Figure 6 Execution Times for Multiple LOS

III. Other Applications

The Radar Altitude and Line of Sight Attachment has provided a solid foundation for applications which go beyond the original concept.

Cockpit Maps.

DTED arrays can be used to generate effective dynamic cockpit and laboratory maps. These are used by research pilots to navigate over simulated terrain, by air combat pilots as tactical situation displays, and by the research staff in the laboratory to keep track of an aircraft's progress over a test track.

Creating a cockpit map from a DTED array is more complicated than creating a static map. There are several technical problems which must be overcome to attain a scalable display with an adequate refresh rate above ten or fifteen Hertz.

Since points in most graphics systems do not scale by definition, filled rectangles must be used instead. Typical DTED arrays in recent research projects have contained roughly one million altitude posts. The display loop which colors and plots this number of rectangles can take five to fifteen seconds even on a modern high-speed graphics workstation.

One solution is to present a static bit-mapped display in a non-rotated "North up" position while moving a rotating cursor (such as a helicopter icon) across it. A small cursor bit map of sixteen points square is drawn last over the static image. In the next frame this sixteen by sixteen point area is refreshed to its original condition from the DTED array before the cursor is plotted again. When larger or more complicated features like waypoint traces and threat symbols are moved, the researcher is asked to incur the 5 to 15 second time penalty required to refresh the entire map image.

Moving Map displays

Some research missions have required high-performance moving maps for displaying several dynamic elements, such as inertially-placed aircraft guidance symbology (tunnels in the sky, for example), and which could be quickly and reliably reprogrammed to meet evolving research needs. In these cases it has proven practical to use an interactive editor for hand-digitizing contours from the DTED array. Double-buffered maps of this kind, drawn from lines, will rotate, translate, and scale at high speeds.

The Generation of Large Sensor Fields

In a recent Terrain Following /Terrain Avoidance project, the technology behind the run-time module was used to construct a forward-looking radar sensor field by

moving the target point around to simulate the search pattern of an actual device and performing repeated LOS calculations from the own-ship eyepoint. The grid of ranges thus built up was then used to augment the static DTED bit map within the Terrain Following /Terrain Avoidance trajectory computer.

IV. Conclusions

This Software has had a Long, Useful Life.

From its inception the concept of the Radar Altimeter and the Line-of-Sight Attachment was intuitively obvious. Subsequently, the software has performed exactly as predicted. The earliest versions were produced on schedule with a small budget.

Because of the simplicity of its individual parts, the clarity of the underlying concepts and the isolation of the Attachment from other elements of the simulation system, the software has been able to undergo a steady program of refinement which has been totally transparent to the user.

The Software has been used in Dozens of Simulations

The Radar Altitude and Line-of-Sight Attachment has now been used in dozens of simulations. Because its performance and resolution can be readily adjusted to accommodate varying computer resources, it has always been able to fulfill its objectives in a completely satisfactory manner. At the same time it has unburdened the image generators, freeing them for the more important primary task of generating out-the-window scenes.

Bibliography

¹ Newman, W. M. and Sproull, R.F.: "Principles of Interactive Computer Graphics," McGraw-Hill Book Company, New York, New York, 1979, pp. 20-22

²Harrington, S.: "Computer Graphics," McGraw-Hill Book Company, New York, New York, 1983.

THE NUMERICAL ERRORS IN INVERSE SIMULATION

K. C. Lin*
University of Central Florida
Orlando, FL 32816
P. Lu**
Iowa State University
Ames, IA 50011
M. Smith†
University of Central Florida
Orlando, FL 32816

Abstract

This paper analyzes the numerical errors of two categories of inverse simulation algorithms: differentiation inverse method and integration inverse method. A first-order differential equation is used to demonstrate the concept. The results of the inverse simulation are used as step-function inputs to the differential equation for a forward simulation. The result is then compared with the original trajectory for the error measure. A second-order differential equation is used to demonstrate the problem which the integration inverse method may encounter when there are unspecified state variables in the problem. A point-mass aircraft model is used as a numerical example for the comparison of these two methods.

Introduction

Simulation, or forward simulation, is a numerical process that, by knowing the inputs of a dynamical system, finds the outputs. Inverse simulation, as its name indicates, is the inverse process of the forward simulation, i.e., by knowing the outputs of a dynamical system, finds the inputs. Inverse simulation has attracted a lot of attention in recent years because of its application in the field of aircraft control.¹⁻⁴ However, only a few papers discuss the numerical procedures of inverse simulation.^{5,6} This paper suggests a procedure to analyze the errors of the inverse simulation algorithms.

There are two major categories of algorithms for inverse simulation: the *differentiation inverse method* and the *integration inverse method*.⁵ The differentiation inverse method uses numerical differentiation to evaluate the time derivatives of the desired trajectory. The controls are then calculated directly

from the differential equations. The integration inverse method assumes that the controls are constant in the step time used to discretize the desired trajectory. An initial guess of the controls is used for the forward simulation. The output variables obtained from the simulation are compared with the variables of the desired trajectory. Based on the errors, the guessed controls are modified using, for example, Newton's method. The process repeats itself until the simulation result converges to the desired trajectory at the end of that step time. Then, it proceeds to the next step time.

Numerical Differentiation

The numerical differentiation formulas are based on the following theorem:⁷

Theorem 1 *If t_0, t_1, \dots, t_n are distinct numbers in the interval $[a, b]$, and if $x \in C^{n+1}[a, b]$, a number $\xi(t)$ in (a, b) exists with*

$$x(t) = \sum_{j=0}^n x(t_j) L_{n,j}(t) + \frac{x^{(n+1)}(\xi(t))}{(n+1)!} \prod_{j=0}^n (t - t_j) \quad (1)$$

where

$$L_{n,j}(t) = \prod_{\substack{i=0 \\ i \neq j}}^n \frac{(t - t_i)}{(t_j - t_i)} \quad (2)$$

The time derivative of $x(t)$ at time t_k , $k = 0, \dots, n$, can be obtained from Eqs. (1) and (2) as:

$$\dot{x}(t_k) = \sum_{j=0}^n x(t_j) \dot{L}_{n,j}(t_k) + \frac{x^{(n+1)}(\xi(t))}{(n+1)!} \prod_{\substack{i=0 \\ i \neq k}}^n (t_k - t_j) \quad (3)$$

When the step times are small, i.e., $|t_k - t_j| \ll 1$, the second term in Eq. (3) can be considered to be the truncation error, and the formula becomes

$$\dot{x}(t_k) \cong \sum_{j=0}^n x(t_j) \dot{L}_{n,j}(t_k) \quad (4)$$

*Assistant Professor of Aerospace Engineering
AIAA member

**Assistant Professor of Aerospace Engineering
AIAA member

†Student Research Assistant

For example, if $n = 1, k = 0$, Eq. (4) becomes

$$\dot{x}(t_0) \cong \frac{x(t_1) - x(t_0)}{t_1 - t_0} \quad (5)$$

Its local truncation error is

$$\epsilon_f = \frac{\ddot{x}(\xi(t))}{2}(t_0 - t_1) \quad (6)$$

This is known as the *forward-difference formula*. On the other hand, if $n = 1, k = 1$, Eq. (4) becomes

$$\dot{x}(t_1) \cong \frac{x(t_1) - x(t_0)}{t_1 - t_0} \quad (7)$$

The local truncation error is

$$\epsilon_b = \frac{\ddot{x}(\xi(t))}{2}(t_1 - t_0) \quad (8)$$

This is known as the *backward-difference formula*.

Both forward- and backward-difference formulas have local truncation errors proportional to the step time ($t_1 - t_0$). Hence, they are formulas of order one.

In the case of $n = 2$, Eq. (4) becomes

$$\begin{aligned} \dot{x}(t_k) &\cong x(t_0) \frac{2t_k - t_1 - t_2}{(t_0 - t_1)(t_0 - t_2)} \\ &+ x(t_1) \frac{2t_k - t_0 - t_2}{(t_1 - t_0)(t_1 - t_2)} \\ &+ x(t_2) \frac{2t_k - t_0 - t_1}{(t_2 - t_0)(t_2 - t_1)} \end{aligned} \quad (9)$$

where $k = 0, 1$, or 2 .

If the step times are all equal, i.e., when

$$x(t_1) = x(t_0) + h \quad \text{and} \quad x(t_2) = x(t_0) + 2h, \quad h \neq 0,$$

Eq. (9) represents the following three formulas:

$$\dot{x}(t_0) \cong \frac{1}{2h}[-3x(t_0) + 4x(t_1) - x(t_2)] \quad (10)$$

$$\dot{x}(t_1) \cong \frac{1}{2h}[-x(t_0) + x(t_2)] \quad (11)$$

$$\dot{x}(t_2) \cong \frac{1}{2h}[x(t_0) - 4x(t_1) + 3x(t_2)] \quad (12)$$

They are forward-, central-, and backward-difference formulas, respectively. Their local truncation errors are

$$\epsilon_f = \frac{x^{(3)}(\xi(t))}{3}h^2 \quad (13)$$

$$\epsilon_c = -\frac{x^{(3)}(\xi(t))}{6}h^2 \quad (14)$$

$$\epsilon_b = \frac{x^{(3)}(\xi(t))}{3}h^2 \quad (15)$$

respectively. They are all formulas of order two.

Formulas for $n = 3$ and above can be derived using a similar procedure.

Error Analysis

The error of a numerical method is usually defined by its local truncation error. This paper suggests another definition for the order of an inverse simulation method.

The error of inverse simulation can be measured by applying the controls back to the differential equations, and comparing the forward simulation result with the desired trajectory. The inverse simulation process, in general, leads to controls in discrete values. The controls can be assumed to be constant (step function) in the step time used to discretize the desired trajectory. The forward simulation uses a small enough integration step such that the integration error is negligible. The error obtained this way is a global error instead of a local error as in the case of truncation error. The order of the inverse simulation method is decided by how this error changes with respect to step time. The following is an example used to demonstrate this concept.

Example 1 A first-order differential equation is given by

$$\dot{x} + x = u(t) \quad (16)$$

with initial condition $x(0) = 0.5$. The desired trajectory is defined by

$$x(t) = 2 - 1.5e^{-t} \quad (17)$$

between $0 \leq t \leq 1$. The exact solution for the inverse simulation is $u(t) = 2$.

The differentiation inverse method is applied to solve this inverse problem. The desired trajectory given by Eq. (17) is first sampled with a constant step time h to obtain a sequence $x(t_k)$, where $t_k = kh$ and k is an integer between 0 and $1/h$. After $\dot{x}(t_k)$ is evaluated numerically using one of the numerical differentiation formulas discussed in the last section, the control can be obtained from the following equation:

$$u(t_k) = \dot{x}(t_k) + x(t_k) \quad (18)$$

Figure 1 shows the control time history obtained by using Eq. (5) to evaluate $\dot{x}(t_k)$, with the step time $h = 0.1$.

To analyze the error of the inverse simulation method, the step-function control is applied to Eq. (16) to run a forward simulation. The simulation result at $t = 1$, the final time, is compared with the desired trajectory at the same time to calculate the error. Figure 2 shows the absolute value of error as a function of step time h , when Eqs. (5) and (7) are used to evaluate $\dot{x}(t_k)$. The straight lines in the figure indicate that both methods are of order one.

Figure 3 shows the absolute value of error as a function of step time h , when Eqs. (10), (11), and

(12) are used to evaluate $\dot{x}(t_k)$. The parabolas in the figure indicate that all three methods are of order two.

The orders of the methods agree with the orders of the formulas used to evaluate $\dot{x}(t_k)$. However, since the errors are global, they provide more information than the local truncation errors. For example, the forward- and backward-difference formulas of the same order have the same absolute local truncation errors. However, from Figures 2 and 3, it is clear that they have different global errors.

Mathematically, as the step time h approaches zero, the numerical differentiation will approach the exact value, and so does the differentiation inverse method. However, the round-off error in digital computers prevents the numerical differentiation from converging to the exact value as h approaches zero.⁷ In practice, proper accuracy can usually be achieved without using a very small step time h .

Another source of error is the uncertainty of the desired trajectory. In the above analysis, the assumption is that the desired trajectory is well defined. However, in some applications, the desired trajectory depends on the on-line measurement of certain variables which may be subject to noises. When there is noise in the desired trajectory, numerical differentiation will produce greater error for using smaller step time h .

Integration Inverse Method

Integration inverse method is an iteration process which eventually finds the controls to make the forward simulation converge to the desired trajectory.⁵ Hence, the global error can be controlled to be as small as desired. Unlike the differentiation inverse method, the global error for the integration inverse method does not increase as the step time h increases. It works even if h is greater than one. Using the definition introduced in the last section, the integration inverse method is a method of order infinity.

For the problems with equal numbers of state variables and controls, such as Example 1, the integration inverse method works for practically all step time h . However, if there is any uncontrolled state variable in the problem, the integration inverse method may have problems for small step time h , as demonstrated in the following example.

Example 2 A second-order differential equation is given by

$$\ddot{x}(t) + \dot{x}(t) + x(t) = u(t) \quad (19)$$

with initial conditions $x(0) = 0$ and $\dot{x}(0) = 0$. The desired trajectory is defined by

$$x(t) = \frac{1}{13} \left\{ e^{-0.5t} [2 \cos \sqrt{0.75}t \right.$$

$$+ (7/\sqrt{0.75}) \sin \sqrt{0.75}t] - 2 \cos 2t - 3 \sin 2t \} \quad (20)$$

between $0 \leq t \leq 10$. The exact solution for the inverse simulation is $u(t) = \sin 2t$.

In this example, there are two state variables, x and \dot{x} , but only one of them, x , is specified by the desired trajectory. When the integration inverse method is used, the uncontrolled variable, \dot{x} , is updated by forward simulation. The integration error, no matter how small it is, will be carried over from one step to the next step, and keep accumulating. Figure 4 shows the results for $h = 0.01$ second. The solution exhibits a high frequency oscillation, which is caused by the uncontrolled state variable, \dot{x} . Smaller step time h can lead to an unstable result. A detailed analysis of this problem can be found in Reference 6.

Numerical Example

In this section, an aircraft trajectory control example is used to demonstrate the use of the above inverse simulation methods.

Example 3 Figure 5 shows the point-mass model of an aircraft in the vertical plane. It is a two-dimensional model. The coordinates used to describe the position of the aircraft are x , the horizontal distance, and h , the altitude. The other two state variables are v , the speed, and γ , the pitching angle. The equations of motion are given by:

$$\dot{x} = v \cos \gamma \quad (21)$$

$$\dot{h} = v \sin \gamma \quad (22)$$

$$\dot{v} = \frac{T - (1/2)\rho v^2 S C_{D0}(1 + \lambda^2)}{m} - g \sin \gamma \quad (23)$$

$$\dot{\gamma} = \frac{(1/2)\rho v S C_L^* \lambda}{m} - \frac{g}{v} \cos \gamma \quad (24)$$

where

$$g = 9.8 \text{ m/sec}^2, \text{ constant,}$$

$$m = 1755 \text{ Kg, mass of the aircraft, constant,}$$

$$S = 17.09 \text{ m}^2, \text{ reference area, constant,}$$

$$\rho = 1.2 \text{ Kg/m}^3, \text{ air density, constant,}$$

$$C_{D0} = 0.0174, \text{ drag coefficient, constant,}$$

$$C_L^* = 0.52, \text{ lift coefficient, constant,}$$

$$T = \text{thrust, control variable,}$$

$$\lambda = \text{control variable.}$$

The desired trajectory is defined by:

$$x = 100t \quad (25)$$

$$h = 2500 + 100(1 - e^{-0.2t}) \quad (26)$$

for $0 \leq t \leq 5$. The objective is to find the time histories of the control variables T and λ .

Figures 6 - 13 show the results of the inverse simulation (step function) compared with the exact solution (smooth curve). Figures 6 and 7 are the results of the differentiation inverse method for $h = 0.5$ second. Figures 8 and 9 are the results of the differentiation inverse method for $h = 0.05$ second. These results show that as the step time h decreases, the accuracy increases.

Figures 10 and 11 are the results of the integration inverse method for $h = 0.5$ second. Figures 12 and 13 are the results of the integration inverse method for $h = 0.05$ second. These results show that as the step time h decreases, the oscillation becomes more severe.

Conclusions

This paper has suggested a procedure to measure the global errors of the inverse simulation methods. This procedure is used to analyze the errors for the differentiation inverse method and the integration inverse method. The comparisons of these two methods can be summarized as:

- The integration inverse method, in general, has better accuracy than the differentiation inverse method.
- When there is an uncontrolled state variable, the integration inverse method may be unstable for small step time h .
- The differentiation inverse method is valid only when the step time h is less than one.
- If the desired trajectory has uncertainty, the noise will be amplified if a very small step time h is used in the differentiation inverse method.
- The differentiation inverse method runs much faster than the integration inverse method.

All the examples in this paper are in the category in which the number of state variables specified by the desired trajectory is equal to the number of control variables. If the numbers are different, the algorithms used to do inverse simulation are more complicated. The error analysis, therefore, needs further study.

References

- 1 Kato, O., and Suguira, I., "An Interpretation of Airplane General Motion and Control as Inverse Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 9, No. 2, 1986, pp. 198-204.
- 2 McKillip, R. M., Jr., and Perri, T. A., "Helicopter Flight Control System Design and Evaluation for NOE Operations Using Controller

Inversion Techniques," *Proceedings of the 45th Annual Forum of the American Helicopter Society*, Alexandria, VA, May, 1989, pp. 669-680.

- 3 Kato, O., "Attitude Projection Method for Analyzing Large-Amplitude Maneuvers," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 1, 1990, pp. 22-29.
- 4 Thompson, D. G., and Bradley, R., "Development and Verification of an Algorithm for Helicopter Inverse Simulation," *Vertica*, Vol. 14, No. 2, 1990, pp. 185-200.
- 5 Hess, R. A., Gao, C., and Wang, S. H., "Generalized Technique for Inverse Simulation Applied to Aircraft Maneuvers," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 5, 1991, pp. 920-926.
- 6 Lin, K. C., "Limitations on the integration inverse methods for Inverse Simulation," *Journal of Guidance, Control, and Dynamics*, to appear.
- 7 Burden, R. L., and Faires, J. D., "Numerical Analysis," 3rd ed., Prindle, Weber & Schmidt, Boston, MA, 1985, pp. 87-145.

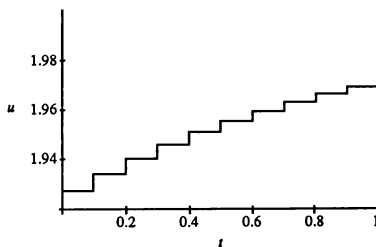


Figure 1: u vs. t , differentiation inverse method, $h = 0.1$ second.

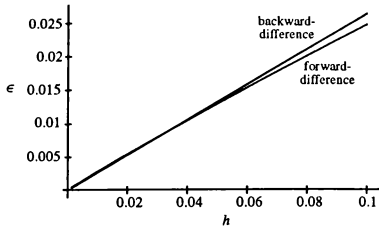


Figure 2: ϵ vs. h , Eqs. (5) and (6).

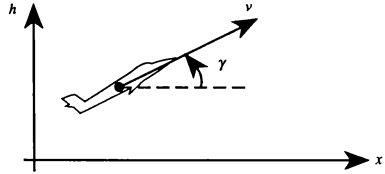


Figure 5: The point mass aircraft model.

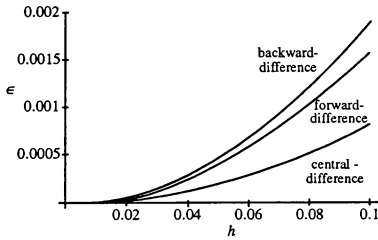


Figure 3: ϵ vs. h , Eqs. (10), (11) and (12).

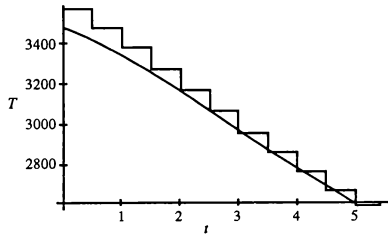


Figure 6: T vs. t , differentiation inverse method, $h = 0.5$ second.

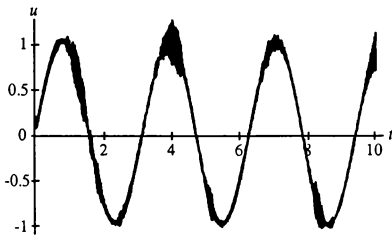


Figure 4: u vs. t , integration inverse method, $h = 0.01$ second.

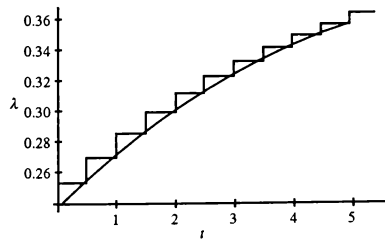


Figure 7: λ vs. t , differentiation inverse method, $h = 0.5$ second.

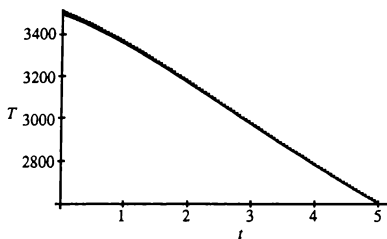


Figure 8: T vs. t , differentiation inverse method, $h = 0.05$ second.

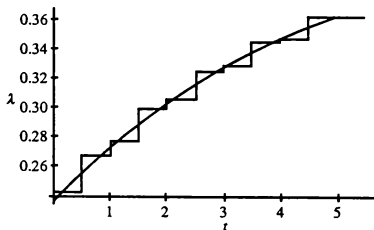


Figure 11: λ vs. t , integration inverse method, $h = 0.5$ second.

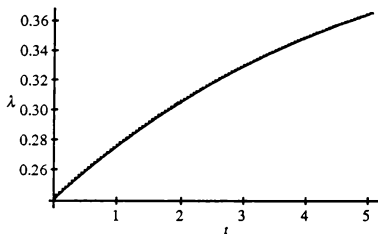


Figure 9: λ vs. t , differentiation inverse method, $h = 0.05$ second.

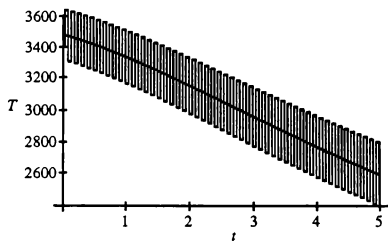


Figure 12: T vs. t , integration inverse method, $h = 0.05$ second.

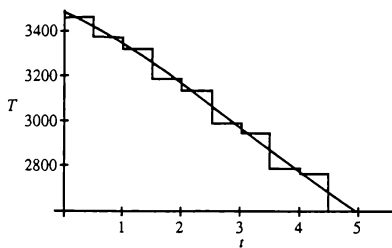


Figure 10: T vs. t , integration inverse method, $h = 0.5$ second.

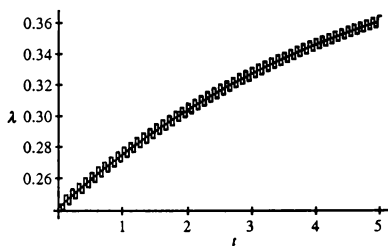


Figure 13: λ vs. t , integration inverse method, $h = 0.05$ second.

A DUAL-EULER METHOD FOR SOLVING ALL-ATTITUDE ANGLES OF THE AIRCRAFT

Huang Xueqiao*
China Flight Test Establishment
Xian China

Abstract

This paper presents a new method, called Dual-Euler Method, for solving exactly all-attitude angles of the aircraft. This method is based on the inverse relationship of both the essence and the singular point-areas between the ordinary and the reversed Euler equations, it separates the area and alternatively calculates their essence areas. That is, it still retains the advantage of straightforwardness and high accuracy of calculation in the essence areas of the ordinary Euler equations, and can utilize the essence areas of the reversed Euler equations instead of the singular areas of the ordinary Euler equations. In this dual-Euler angle coordinate system, the essence area is expanded to the whole area and the singular area can be eliminated. Without principle defects and method errors, so the dual-Euler method can fully overcome the singularity and get the most exact all-attitude angles of the aircraft. The testing results show that the transit of the attitude angle in the whole area with the dual-Euler method is smooth and bona fide, thereby it is an ideal all-attitude equation.

List of Symbols

A transformation matrix from Earth axes to body axes, also called attitude matrix
a._{ij} elements (i row and j rank) of A
[a]_i = [a] transformation matrix of rotation about i axis, as:

$$[\Phi]_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Phi & \sin\Phi \\ 0 & -\sin\Phi & \cos\Phi \end{bmatrix}$$

$$[\Theta]_y = \begin{bmatrix} \cos\Theta & 0 & -\sin\Theta \\ 0 & 1 & 0 \\ \sin\Theta & 0 & \cos\Theta \end{bmatrix}$$

$$[\Psi]_z = \begin{bmatrix} \cos\Psi & \sin\Psi & 0 \\ -\sin\Psi & \cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

[a]^T transpose matrix of [a]
Φ, Θ, Ψ three attitude angles, roll angle, pitch angle and heading angle in ordinary Euler angle system
Φ_r, Θ_r, Ψ_r three rotating angles in reversed Euler angle system
P, Q, R three aircraft rotating rates about X, Y and Z axis

1. Introduction

Attitude angles of the aircraft are also called the Euler angles. The angles in flight simulation can be usually solved by a single set of the Euler equations. This is a simple and old method but it is the most exact and feasible method in general.

The old method has difficulty in solving the all-attitude simulation and serious effects on solution accuracy due to natural defect of the singularity in the Euler angle system. Though it is possible to solve all-attitude simulation by use of a dead band fixed-value method, it can not fully remove the errors of the singular areas so that the classical equations of the single-Euler method is very difficult to suit for solving all-attitude of the aircraft.

In the past, there were many substitute methods to overcome the difficulty in calculating in the neighborhood of the singular points. One of the substitute methods, the quaternion method, used to solve all-attitude simulation in nine cases out of ten. But the quaternion method is not very perfect. Besides leading into simplified hypothesis and conditions, there are frequent phenomena out of the boundary owing to integrating to get parameters of field of definitions. Due to all these factors of principle defect, the quaternion method will produce method errors. After the correcting process, it still can not fully remove the errors and only gets an approximate solution. Other methods are not better than the quaternion method. So there are defects in all these methods.

In order to fully overcome the singularity, in 1978, a dual-set of the Euler equations was presented by use of separating the Euler angle system into two kind of areas with new the method of the alternating calculation. It is cal-

*Senior Engineer

led a dual-Euler method. Due to the concept simplicity, straightforwardness, convenience in application it is an ideal all-attitude equations.

2. The Ordinary Euler Angle System

According to concepts of the flight dynamics, the attitude angles Ψ - Θ - Φ (i.e. the ordinary Euler angles) round the aircraft body axes Z-Y-X relative to the Earth axes and arrive at the new situation. The relationship between their angle velocities, and P, Q, and R rotating rates of the body axes can be described by the (ordinary) Euler equations, namely

$$\left. \begin{aligned} \dot{\Phi} &= P + \Psi \sin \Theta \\ \dot{\Theta} &= Q \cos \Phi - R \sin \Phi \\ \dot{\Psi} &= (Q \sin \Phi + R \cos \Phi) / \cos \Theta \end{aligned} \right\} \quad (1)$$

The above equations are integrated and given Φ and $\Psi \leq 180^\circ$ field of definitions, and then all trigonometric functions are converted. Thereby the aircraft attitude angles, and their sine and cosine functions can be directly resolved.

$\Theta = \pm 90^\circ$ is a pair of singular points in the above equations. In this case, both Φ and Ψ are going to infinity so that the attitude angles can not be defined. It is shown that the ordinary Euler equations will produce solution errors in the neighborhood of the singular points so that only use of the ordinary Euler equations is too difficult to calculate for all attitude angles.

3. The Reversed Euler Angle System

Supposing the reversed Euler angle system is that attitude angles Ψ_r - Φ_r - Θ_r round the aircraft body axes Z-X-Y relative to the Earth axes and arrive at the new situation. The relationship formulae between their angle velocities, and P, Q, and R rotating rates of the body axes calls the reversed Euler equations, namely

$$\left. \begin{aligned} \dot{\Phi}_r &= P \cos \Theta_r + R \sin \Theta_r \\ &= (P \cos \Theta \cos \Phi + R \sin \Theta) / \sqrt{1 - \cos^2 \Theta \sin^2 \Phi} \\ \dot{\Theta}_r &= Q - \dot{\Phi}_r \sin \Theta_r = Q - \dot{\Psi}_r \cos \Theta \sin \Phi \\ \dot{\Psi}_r &= (-P \sin \Theta_r + R \cos \Theta_r) / \cos \Phi_r \\ &= (-P \sin \Theta + R \cos \Theta \cos \Phi) / (1 - \cos^2 \Theta \sin^2 \Phi) \end{aligned} \right\} \quad (2)$$

The above equations are integrated and given $\Psi_r \leq 180^\circ$ field of definitions, and then all trigonometric functions are converted. Thereby the reversed Euler angles, and their sine and cosine function can be resolved. From above equations, we can know that there is also a pair of singular points in $\Phi_r = \pm 90^\circ$. In this case, $\dot{\Theta}_r$ and $\dot{\Psi}_r$ are going to infinity so that the reversed Euler angles can not be defined. The reversed Euler equations can also not be used for solving aircraft all-attitude angles.

4. Coordinate Transformation of Two Angles System

4.1 Coordinate Transformation of Direction Cosines Attitude Matrix

The direction cosine expressions of aircraft body axes around the ordinary and the reversed Euler angles system relative to the Earth axes are as follows:

$$\begin{aligned} A &= [\Phi]_x [\Theta]_y [\Psi]_z \\ &= [\Theta]_y [\Phi]_x [\Psi_r]_z \end{aligned} \quad (3)$$

Spreading out the above formula to get the direction cosine matrix expressions:

$$\begin{aligned} &\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \\ &= \begin{bmatrix} \cos \Theta \cos \Psi & \cos \Theta \sin \Psi & -\sin \Theta \\ -\cos \Phi \sin \Psi & \cos \Phi \cos \Psi \\ +\sin \Theta \sin \Phi & +\sin \Theta \sin \Phi & \cos \Theta \sin \Phi \\ \times \cos \Psi & \times \sin \Psi \\ \sin \Phi \sin \Psi & -\sin \Phi \cos \Psi \\ +\sin \Theta \cos \Phi & +\sin \Theta \cos \Phi & \cos \Theta \cos \Phi \\ \times \cos \Psi & \times \sin \Psi \end{bmatrix} \\ &= \begin{bmatrix} \cos \Theta_r \cos \Psi_r & \cos \Theta_r \sin \Psi_r & -\sin \Theta_r \cos \Phi_r \\ -\sin \Theta_r \sin \Phi_r & +\sin \Theta_r \sin \Phi_r & -\sin \Theta_r \cos \Phi_r \\ \times \sin \Psi_r & \times \cos \Psi_r \\ -\cos \Phi_r \sin \Psi_r & \cos \Phi_r \cos \Psi_r & \sin \Phi_r \\ \sin \Theta_r \cos \Psi_r & \sin \Theta_r \sin \Psi_r \\ +\cos \Theta_r \sin \Phi_r & -\cos \Theta_r \sin \Phi_r & \cos \Theta_r \cos \Phi_r \\ \times \sin \Psi_r & \times \cos \Psi_r \end{bmatrix} \end{aligned} \quad (4)$$

Formular (3) multiplied by $[\Phi]'$ and $[\Theta]'$ on the left-hand side of it and spreaded out, we may get the correlative matrix formula.

From above formulas we may find sine and cosine interactive equations of the ordinary and the reversed Euler angles.

4.2 Conversion of Ordinary / Reversed Euler Angles System

When $|\Theta| > 30^\circ$:

$$\left. \begin{aligned} \sin\Phi_r &= a_{23} = \cos\Theta \sin\Phi \\ \cos\Phi_r &= \sqrt{1 - a_{23}^2} \\ \Phi_r &= \sin^{-1} a_{23} \\ \sin\Theta_r &= -a_{13} / \cos\Phi_r \\ \cos\Theta_r &= a_{33} / \cos\Phi_r \\ \Theta_r &= \tan^{-1}(-a_{13} / a_{33}) \\ \Psi_r &= \tan^{-1}(-a_{21} / a_{22}) \end{aligned} \right\} \quad (5)$$

4.3 Conversion of Reversed / Ordinary Euler Angles System

$$\left. \begin{aligned} \sin\Theta &= -a_{13} = \sin\Theta_r \cos\Phi_r \\ \cos\Theta &= \sqrt{1 - a_{13}^2} \\ \Theta &= \sin^{-1} -a_{13} \\ \sin\Phi &= a_{23} / \cos\Theta \\ \cos\Phi &= a_{33} / \cos\Theta \quad \left. \begin{aligned} &\text{if } \cos\Theta \neq 0 \\ &\Phi = \tan^{-1}(a_{23} / a_{33}) \end{aligned} \right\} \quad (6) \\ \sin\Psi &= -a_{21} \cos\Phi + a_{31} \sin\Phi \\ \cos\Psi &= a_{22} \cos\Phi - a_{32} \sin\Phi \\ \Psi &= \tan^{-1}(a_{12} / a_{11}) \end{aligned} \right\}$$

5. Singularity Analysis

The results of theoretical analysis showed that there are a pair of singular points which can not be defined and make the errors of the calculation, and a pair of essence points which can produce the most exact solution in all Euler angle systems. The two pairs of those points are 90° out of phase and perpendicular to each other.

If we let a circular angle to divide into four areas of $\pm 45^\circ$. Two of them are the es-

sence areas in $\pm 45^\circ$ about the essence points in which the exact solution can be got. The other two are the singular areas with errors. Its error is a gradual-change process of nonlinearity from quantity to quality.

From formula (1): the singular points of the ordinary Euler equations are $\Theta = \pm 90^\circ$, while the essence points of them are $\Theta = 0^\circ$ (and 180°). From formula (2): the singular points of the reversed Euler equations are $\Theta = 0^\circ$ (and 180°) and the essence points of them are in $\Theta = \pm 90^\circ$. It shows that the singular and the essence points between the ordinary and the reversed Euler system are the inverse relationship, i.e. the point-areas about $\Theta = 0^\circ$ (and 180°) of the level flight coincide with both point-areas of the essence of ordinary Euler equations and of the singular of reversed Euler equations, and the point-areas about $\Theta = \pm 90^\circ$ of the vertical flight coincide with both point-areas of the singular of ordinary and of the essence of reversed Euler equations. (as seen in Fig. 1).

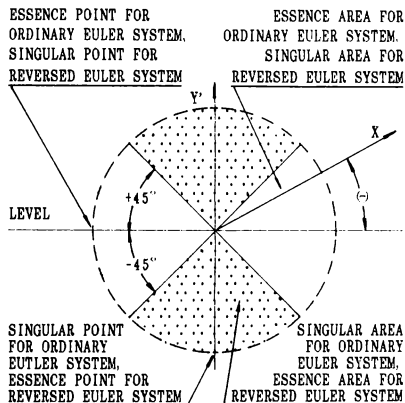


Fig. 1. Area Distribution for Two Angle systems

6. Principle of Dual-Euler Method

In order to fully overcome the singularity of the classical single-Euler equations and to solve the exact all-attitude angles of the aircraft, the dual-Euler method based on the inverse relationship between the singularity of both the

reversed and the ordinary Euler equations with divided area. That is, when $|\Theta| \leq 45^\circ$ (and $< 135^\circ$) in general, we may directly solve the ordinary Euler equations as same as the single-Euler equations do to keep advantages of the concept simplicity, straightforwardness, and high accuracy in the essence area. When $|\Theta| > 45^\circ$ (and $< 135^\circ$) in non-normal, in order to escape the singularity difficulty of solving the attitude velocities in the singular areas of the ordinary Euler equations, we solve the reversed Euler equations so as to use its essence area to get accuracy resolutions instead of the ordinary Euler equations. With this method, the essence area of the Euler angle system is expanded to a whole area and the singular areas are eliminated, and without principle defects and method

errors. So the dual-Euler angle coordinate systems can fully overcome the singularities of the Euler equations and get the most exact attitude angles of the aircraft as same as the essence areas of the classical single Euler equations.

The operation flow chart of the dual-Euler method is shown in Fig 2. Besides solving the ordinary or the reversed Euler equations, it includes all the elements of the attitude, and coordinate transformation of the initial values required in the next alternating process and of the aircraft attitude parameters required finally. If this resolution is arranged in a reasonable way, we could get almost same sampling iterate of cycle time for the ordinary or the inversed Euler equations.

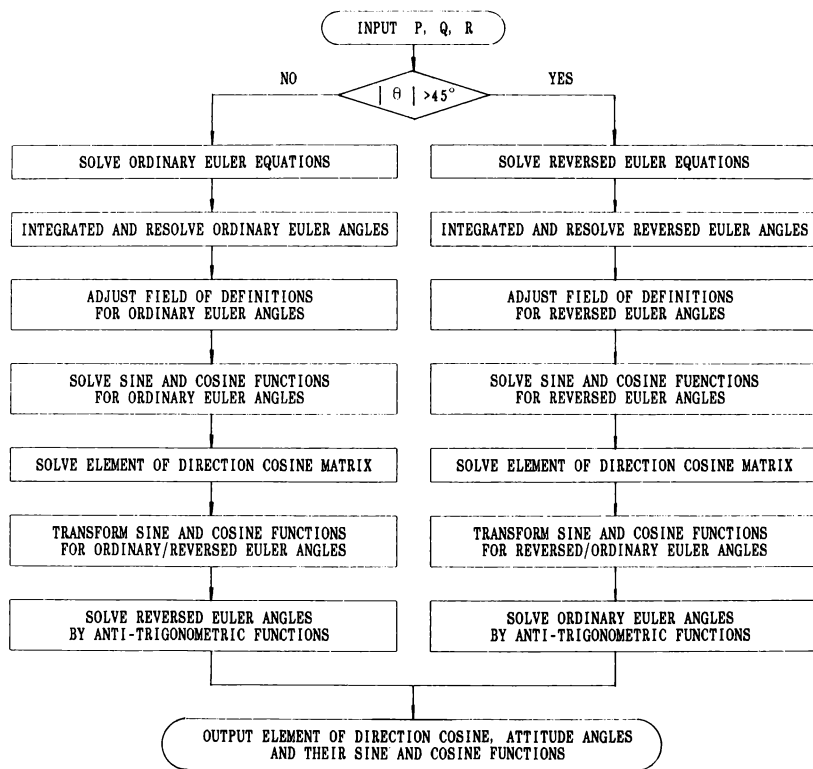


Fig. 2. Flow Chart of Dual-Euler Method Solution

7. Experimental Verification

In order to verify the correctness of the dual-Euler method, we compared the dual-Euler method with the single-Euler and the quaternion methods through a good many of numerical-value simulations and application testings of flight simulation.

The numerical-value simulation tested mainly in loop and roll flight maneuverings. Their math models for the input cues are:

for loop

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} \pi/4 \sin \pi t \\ \pi \\ -P \end{bmatrix} \quad (7)$$

for roll

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} \pi \\ \pi/4 \sin \pi t \\ -Q \end{bmatrix} \quad (8)$$

the solution step is 0.05 sec.

The test results show that:

(a) The dual-Euler method and the quaternion method can overcome the singularity and the transition of attitude angles are smooth in whole area, but during continual loop or roll flights the test results of attitude angles are very different between two methods. It means that at least the result's accuracy for one of them is not high. The single-Euler method still has a vibration phenomenon of the attitude angles by use of a dead band fixed-value in the singular area (in the loop flight), so it does not fully overcome the singularities and its accuracy is not high in the singular areas.

(b) During continual looping or rolling flights solving at 10000 times cycles, the dual-Euler method did not overstep the boundary due to principle defects. The quaternion method, after large error correction, still has frequently the phenomena of out-of-boundary with more than 1500 times for the limit defined parameters. If let $P > \pi$ (the $P_{\max} > 1.5\pi \sim 2\pi$ for a modern high-speed aircraft), times and values of out-of-boundary will increase greatly. The test results show that we can get the most exact solution for the dual-Euler method while the approximate solution for the quaternion method. These differences mentioned above between two methods are method errors due to the principle defects with the quaternion method, so that the

potentiality for eliminating the singularities of the dual-Euler method has been verified.

(c) Flights in the man-machine loop on the simulator with the dual-Euler method have achieved good results including the indication of instruments and the scene of Sky-Earth image. Specially flights of the loop up to vertical attitude in the singular area are very true. The fact of the matter is that utility of the dual-Euler method is sure enough.

(d) If say defects of the dual-Euler method, the method has a longer cycle time for the solution than other methods (more a factor of 1.5 than the single-Euler method and more a factor of 1.1 than the quaternion method). As viewed from fully overcoming the singularity, it is worth at the cost of a little more cycle time.

The correctness of theory analysis have been fully proved by the test results. Other method is incomparable with it. No body will put a question to creditability and applicability for the dual-Euler method.

8. Fields of Application

The Dual-Euler method can not only solve exactly all-attitude angles in flight simulation to enhance simulation fidelity, but on the principle of the method, it can also produce the most exact solution of all-attitude angles for the vehicle. To date, for all the vehicle (including aircraft, spacecraft, satellite, missile and rocket and so on) all attitude sensors in flight measuring test systems, flight instruments, flight control systems, fire control systems, inertial navigation and autopilots have a phenomenon of gimbal lock in gyro platforms. It represents the singularity of this Euler angle system in a physical construction. The singularity has serious influence on exactly measuring, display and control and can not be entirely overcome upto now. If the principle construction of attitude angle gyro sensors can be improved by the theory of the dual-Euler method, and the bad effect of gimbal lock can be eliminated, the operating dynamic errors of platform-type sensors and the solving method errors of strapdown type sensors can be removed. So the dual-Euler method is a valuable method for solving the difficult problem of the singularity in math and physical link of Euler angle system.

The following measures can be employed:

For platform-type gyroscope, it contains two gyros. One gyro is longitudinal-mounted to measure the ordinary Euler angles, the other is lateral-mounted to measure the reversed Euler

angles. It first gives coordination and conversion of coordinate transform with divided area and then may gain terminal values.

For strapdown-type gyroscope, the dual-Euler method equations can just be used.

9. Conclusion

The dual-Euler method can fully overcome the singularities of the Euler equations and gets the most exact solution in all-attitude. The other kind of method is incomparable with it. Its concept simplicity and the application is convenience. It is an ideal all-attitude equations. The results of theoretical analysis and validation tests showed that the dual-Euler method is feasible and its principles are valuable for reference in aviation and other physical and engineering areas.

References

1. Alfred C. Robinson: On the Use of Quaternions in Simulation of Rigid-Body Motion, WADC TR 58-17 DEC. 1958, or PB 161,732.
2. Greenwood, D. R.: An Extended Euler Angle Coordinate System for Use with All-Attitude Aircraft Simulators, WADD TR 60-372 Aug. 1960, or PB 171,404.

INITIAL DEVELOPMENT OF A RESEARCH FLIGHT SIMULATOR SOFTWARE

T. Caleffi, Eng. and E.M. Belo, Prof.
Department of Mechanical Engineering
São Carlos Engineering School - University of São Paulo
São Carlos - Brazil

Abstract

In this work, steps that were taken to design part of a flight simulator software are described. It comprises the integration, look-up and set of flight differential equations routines and the aircraft data. The software was made in a modular form, to allow future implementations. The method chosen to make the equations integration was the fourth-order Runge-Kutta. The data were disposed in a bidimensional matrix, called data-pool, and the language used was the C language, due to its portability and data structures.

I. Introduction

Flight simulation is an important piece in the aircraft design process. Simulation can be done inside an actual airplane with variable stability, called in-flight simulator, or on the ground, using the well known ground-based simulators.

A typical ground-based simulator usually consists of the real-time digital computer, the cockpit and the visual and motion systems. The computer drives the simulator; the cockpit, normally, imitates the actual aircraft; and the visual and motion systems provide the pilot with the real world scenery and the sensation of movement.

In this work, steps that were taken to design part of a flight simulator software, that commands the computer will be described. In other words, the set of non-linear flight differential equations, solved using numerical solutions, will be presented.

The software was made in a modular form, the data were disposed in a matrix, called data-pool, and the language chosen was the C language, due to its portability and data structures.

It is necessary to emphasized that the work is still being done, in order to develop some tests.

II. Mathematical model

The flight equations were developed from the basic theory of aircraft stability and control, and aerodynamic concepts (Etkin, 1959; Etkin, 1972; Blakelock, 1965; Babister, 1961). At this

stage, it was assumed that the Earth is flat, the gravity is a radial vector, the atmosphere is at rest relative to the Earth and the airplane possesses a plane of symmetry, what means that the inertia with respect to the fore-and-aft plan is zero. In this way, the equations of motion, considering the aircraft as a rigid body flying in steady air, are (see Figure 1):

$$L = I_x \ddot{\phi} - I_{xx}(\dot{\phi} + p\dot{q}) - (I_y - I_z) q \dot{r} \quad (1)$$

$$M = I_y \ddot{q} - I_{xx}(r^2 - p^2) - (I_z - I_x) x \dot{p} \quad (2)$$

$$N = I_z \ddot{r} - I_{xx}(\dot{p} - q\dot{r}) - (I_x - I_y) p \dot{q} \quad (3)$$

$$X = m(\dot{u} + q\dot{w} - rv) + mg \sin \theta \quad (4)$$

$$Y = m(\dot{v} + r\dot{u} - pw) - mg \cos \theta \sin \phi \quad (5)$$

$$Z = m(\dot{w} + p\dot{v} - qu) - mg \cos \theta \cos \phi \quad (6)$$

The solution of this set of equations must be in real-time, due to the presence of the man in-the-loop, who must be supplied with information in a very short time, like in an actual aircraft. In this type of simulation, real-time means a time interval between 20 and 50 milliseconds.

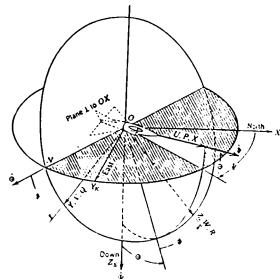


Figure 1 - Aircraft orientation

III. Real-time calculation computer system

Flight simulators must run on a real-time basis because they are manipulated by a pilot to evaluate flight performance and aircraft stability. Presently, digital computers are used to do this. The goal of this work is to develop a software able to solve the non-linear set of flight differential equations, using the C language. Thinking of future implementation, this software was made as modular as possible. Basically, it will comprise three modules, to know:

- integration and look-up routines;
- data-pool; and
- set of flight equations routines.

Look-up routine

The theory of table look-up gives a table of values of one variable Y as a function of a set of values of X. The coordinates, when plotted, look like Figure 2 and we can find the Y value for any general X value. To do this, we have to interpolate adjacent pairs of coordinates, as follows:

$$X_2 \leq X \leq X_3 \Rightarrow Y = Y_2 + \left(\frac{Y_3 - Y_2}{X_3 - X_2} \right) (X - X_2) \quad (7)$$

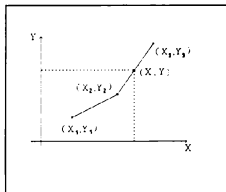


Figure 2 - Interpolation of $Y = f(X)$

Basically, a simple algorithm for this table look-up process has to perform the following steps:

1. to decide between which pair of X values in the table the current X value lies;
2. to calculate the local slope $(Y_3 - Y_2) / (X_3 - X_2)$; and
3. to apply the interpolation formula.

The look-up process chosen to do this work is that published by Sinnott et al. (1990). The method executes a top-down search only during the initialization step. From that on, it returns to the same independent variable intervals as from the previous step, where the independent variable will still most likely reside. If the interval in which one of the independent variable was located has changed in one of the n dimensions, this algorithm calculates the velocity of that independent variable through the table and proceeds directly to the correct new interval.

The advantage of using the algorithm described above is that the table search time is reduced, due to the fact that the top-down search is performed only in the initialization step, after that, the algorithm returns to the interval where the value will most likely reside.

Integration routine

To solve the non-linear set of differential equations of flight, the numerical method chosen was the fourth-order Runge-Kutta for non-linear systems of differential equations.

The Runge-Kutta methods propagate a solution over an interval by combining the information from several Euler-style steps, and then using the information obtained to match a Taylor series expansion up to some higher order.

The classical fourth-order formula is:

$$y_{i+1} = y_i + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4) \quad (8)$$

where:

$$\begin{aligned} k_1 &= f(x_i, y_i) \\ k_2 &= f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_1\right) \\ k_3 &= f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_2\right) \\ k_4 &= f(x_i + h, y_i + hk_3) \end{aligned} \quad (9)$$

Set of flight equations routine

Basically, the modular structure of the program consists of a set of three routines: EQUATIONS, ENGINE and OTHERS.

The EQUATIONS routine computes the forces and moments acting in the aircraft during flight, given the velocities, rates, etc. The ENGINE routine

computes the forces and moments that result from the engine of the aircraft. The OTHERS routine calculates all other parameters necessary to the simulation, considering the limitations of this work.

Data-pool

During the simulation time, all parameters will be stored in a matrix, called data-pool. This was done to reduce the time that the program takes to transfer the parameters from one equation to another.

The data-pool was built as a bidimensional matrix, using pointers to indicate each variable or constant used during the simulation time. Considering that the simulation is independent of the aircraft being simulated, inside certain limits, we can change the aircraft type, replacing the data-pool parameters.

IV. Aircraft orientation

The orientation of any reference frame related to other, can be described by three angles, which define successive rotations about x , y and z axes. In flight dynamics, the angles used are called Euler angles, and are showed in Figure 4.

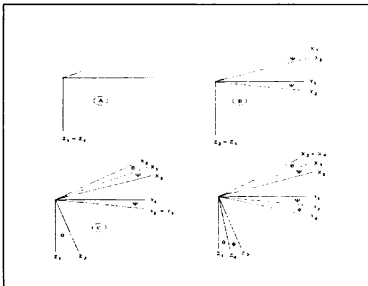


Figure 4 - Successive rotations about the reference frame, to show the Euler angles.

- where: - A = initial situation;
- B = rotation about the Z axis;
- C = rotation about the Y₁ axis;
- D = rotation about the X₂ axis;
- θ = pitch angle;
- ϕ = bank angle; and
- Ψ = azimuth angle.

The problem of Euler angles is the singularities that happen out of the following limits:

$$-\pi \leq \Psi < \pi \quad 0 \leq \Psi < 2\pi \quad (10)$$

$$-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2} \quad (11)$$

$$-\pi \leq \phi < \pi \quad 0 \leq \phi < 2\pi \quad (12)$$

In this work, in order to avoid those singularities, we will use the quaternions method (Robinson, 1958; Mitchell & Rogers, 1965 cited by Rolfe & Staples, 1989) to define the aircraft orientation.

Lets us to consider a frame of axes Oxyz, that must be coincident with the reference frame $Ox_0y_0z_0$, making a simple rotation D, with A, B, C angles. This method consists of a change of variables from the original parameters (A, B, C and D) to a set of quaternion parameters e_0 , e_1 , e_2 and e_3 , as follows:

$$e_0 = \cos \frac{1}{2}D \quad (13)$$

$$e_1 = \cos A \sin \frac{1}{2}D \quad (14)$$

$$e_2 = \cos B \sin \frac{1}{2}D \quad (15)$$

$$e_3 = \cos C \sin \frac{1}{2}D \quad (16)$$

So, it can be shown that the matrix transformation that relates (x, y, z) to (x_0, y_0, z_0) in the A, B, C and D parameters can be substitute by the quaternions using the equations (13), (14), (15) and (16), resulting in the following matrix:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} e_0^2 + e_1^2 - e_2^2 - e_3^2 & 2(e_1e_2 + e_0e_3) & 2(e_1e_3 - e_0e_2) \\ 2(e_1e_2 - e_0e_3) & e_0^2 - e_1^2 + e_2^2 - e_3^2 & 2(e_2e_3 + e_0e_1) \\ 2(e_1e_3 + e_0e_2) & 2(e_2e_3 - e_0e_1) & e_0^2 - e_1^2 - e_2^2 + e_3^2 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \quad (17)$$

Relating the equation (17) to the direction cosines, it can be shown that:

$$e_0 = -\frac{1}{2}(e_1p + e_2q + e_3r) \quad (18)$$

$$\dot{e}_1 = \frac{1}{2} (e_0 p + e_2 r - e_3 q) \quad (19)$$

$$\dot{e}_2 = \frac{1}{2} (e_0 p + e_2 r - e_3 q) \quad (20)$$

$$\dot{e}_3 = \frac{1}{2} (e_0 r + e_1 q - e_2 p) \quad (21)$$

So, in this way, from the velocities p , q and r , we can easily calculate the angles ϕ , Ψ and θ .

V. Tests

To test the software, the chosen method could be the part-task technique, presented by Sizer (1987), where the use of cockpit hardware is not necessary.

This method is called off-line simulation and the software is executed with the single model isolated for debugging, acting as the nucleus for the test configuration. It consists of a three packages to make up the software configuration: control, command and display softwares.

VI. Conclusions

The goal of this work is to show the steps that were taken to develop a initial research flight simulator software. At this stage, we can see that the methods chosen worked well.

The quaternion parameters method did prove its feasibility, showing to be the best way to calculate the angles ϕ , Ψ and θ .

VII. References

- .Babister, A.W. "Aircraft stability and control", Pergamon Press, 1961, 700p.
- .Blakelock, J.H. "Automatic control of aircraft and missiles", John Wiley & Sons, 1965, 348p.
- .Etkin, B. "Dynamics of flight: stability and control", John Wiley & Sons, 1959, 518p.
- .Etkin, B. "Dynamics of atmospheric flight", John Wiley & sons, 1972, 578p.
- .Rolfe, J.M. and Staples, K.J. "Flight Simulation", Cambridge, Press Syndicate of the University of Cambridge, 1989, 282p.
- .Sinnott, M.K., Steck, J.E., Selbert, B.P. and Oetting, R.B. "Alternate table look-up routine for real-time digital flight simulation", Journal of Aircraft, 27 (17): 471-473, may 1990.

.Sizer, A.M. "Part-task simulation techniques", Conference on flight simulation techniques of the American Institute of Astronautics and Aeronautics, Monterey, 1987. *Proceedings*. p.17-23.

ENHANCING REAL-TIME FLIGHT SIMULATION EXECUTION BY INTERCEPTING RUN-TIME LIBRARY CALLS

Namejs Reinbachs
Senior Systems Programmer
SYRE, a division of SYSCON Services, Inc.
NASA Ames Research Center - Moffett Field, California

Abstract

Standard operating system input-output (I/O) procedures impose a large time penalty on real-time program execution. These procedures are generally invoked by way of Run-Time Library (RTL) calls. To reduce the time penalty, as well as add flexibility, a technique has been developed to dynamically intercept these calls. The design and implementation of this technique, as applied to FORTRAN WRITE statements, are described. Measured performance gains using this RTL intercept technique are on the order of 1000%.

Introduction

Frequently, software is required to run under tight timing constraints in order to perform all the necessary operations of flight simulation within an allotted amount of time. In a real-time environment it is imperative that certain tasks complete within time constraints (time-critical), while other tasks have less stringent requirements (non-time-critical). Intercepting the default RTL calls of processes allows greater flexibility in handling these tasks. Operations can be deferred, modified, or even dismissed completely. As a result, the execution of the time-critical sections is expedited significantly.

This work focuses on the design and implementation issues of using this technique within the VAX/VMS architecture. It describes a highly flexible and efficient mechanism which has minimal impact on the operating system. The performance is also evaluated. Design decisions, their motivations, and an RTL overview are presented. Following a discussion of the design and its implementation, a case study is used to demonstrate this technique. Finally, performance measurements are evaluated, and a discussion of the advantages and disadvantages are presented.

Motivation

At NASA Ames Flight Simulation Laboratory (SimLab), real-time aircraft model applications run under a general purpose operating system, and are written entirely in FORTRAN. While this structure provides an excellent development environment, it is not ideal for real-time use. For example, VAX/VMS FORTRAN does not allow for asynchronous input-output operations. If a FORTRAN WRITE statement is executed, the process is stalled until the WRITE request completes. If output is directed to a physical device, the delay may easily become excessive.

The typical method used to solve this problem is to have a special purpose WRITE routine that circumvents the above-mentioned overhead. This, however, necessitates that the FORTRAN programmer learn and remember the new call routines to substitute for more familiar standard calls.

To eliminate this problem an attempt was made to link in replacement routines for the system's FORTRAN WRITE routines. Other than annoying linker warnings, this method seemed to work well. Upon closer examination, however, other problems began to emerge. Internal memory writes within the application required access to the system's routines instead of the replacement routines. Both sets of routines had to be called dynamically, and normal link libraries would not allow this.

Hence, we investigated the use of Run-Time Libraries. Since in the SimLab real-time environment the output from WRITE statements to devices is not needed in real-time, it was decided that this would be an ideal situation in which to intercept the relevant RTL calls. This method enhances real-time performance of flight simulation software as well as allows execution of the original library routines.

RTL Overview

Most libraries are *link libraries* where the desired routines are included in a program at link time. If other programs also need the same routines they receive their own copy. Figure 1 illustrates a simple program that calls routine 'a' twice and routine 'b' once using a normal link library. Note that all references to routines 'a' and 'b' are resolved before the program is loaded.

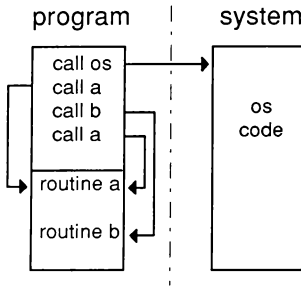


Figure 1. Program with Link Library

By comparison, Run-Time Libraries (RTLs) are not linked with program files but are included at execution time, also called *run-time*. When RTLs are used, only one copy of the library is needed regardless of the number of programs that invoke these routines. This results in reduced disk storage and faster program loading.

Since an RTL is not directly connected to a program, the program must signal the operating system that it has a request for a library routine. This signaling can take place via calls, traps, exceptions, or interrupts.¹ Thus an RTL may be viewed as an extension of the operating system.

Figure 2 illustrates the previous simple program example (figure 1); this time, however, routines 'a' and 'b' are placed in an RTL. Instead of calling the routines directly, they are called indirectly through a dispatch or jump table (referred to as *rtl a* and *rtl b* in the figure).

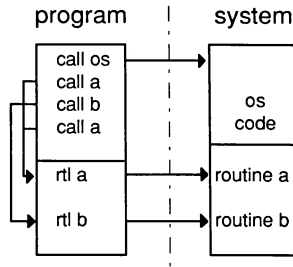


Figure 2. Program with RTL

RTLs assist in the interception of these calls by allowing the addresses of the RTL routines to be substituted with the addresses of user-defined routines. These substitutions can be made dynamically and repeatedly. Figure 3 illustrates the same simple program as figure 2, with user defined routines 'c' and 'd' intercepting the system routines 'a' and 'b' via address substitution in the jump table.

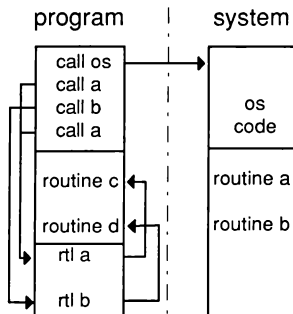


Figure 3. Program with Intercepted RTL

In addition, if the original system routines' addresses are saved before the substitutions are made, these system routines can be invoked if desired. Figure 4 illustrates the same program as figure 3, with user defined routines 'c' and 'd' making calls to the original system RTL routines 'a' and 'b'.

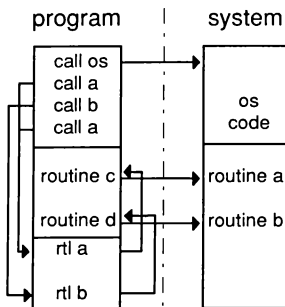


Figure 4. Program with Intercepted and Original RTL

The concept of intercepting an RTL call is analogous to that of error handling (exception handling). Typically, an application will abort if a fatal error is detected (illegal address or divide by zero, for example). As an alternative, programs can be made capable of fielding these errors and performing corrective actions to allow the application to continue. The program may choose to ignore the error, or call a special user supplied routine if a particular error condition is detected. Although the methods of setting up such error handlers are machine specific, the concepts involved are similar.

Intercepting RTL calls extends the concepts of the above mentioned error handling example. Instead of only error conditions, any RTL call

can be intercepted and an alternate routine executed. Or, as with error handling, the normal system routines may be invoked to handle the RTL request.

Design and Implementation

The VAX/VMS implementation of RTLs (referred to as sharable images² under VMS) follows a standard method used by many operating systems. A program linked under version 3 or later of VMS includes a section called the *fixup vector table*.³ This table contains an entry for every RTL routine referenced within the program.

All RTL references within the program are changed to a deferred addressing mode by the linker. Instead of an absolute address, this is the relative address of the proper RTL entry within the fixup vector table. At runtime, this entry will contain the correct virtual address of the target RTL routine. Note that there is only one entry per RTL routine within the fixup vector table regardless of the number of references contained in the program. All references to the same routine point to the same table entry.

In modifying the behavior of RTL routines there exist three addresses of importance:

1. address of the substituted routine,
2. address of the current RTL routine, and
3. address of the corresponding fixup vector.

The following code is a macro that is used to save all three addresses into an intercept table data structure:

```

; -----
;   Intercept Dispatch Table (function & fixup vector locations)
; -----
.macro      INTERCEPT_TABLE  ROUTINE
nameofs    =    %locate(<$>,ROUTINE) + 1
namelen    =    %length(ROUTINE) - nameofs
.psect     INTERCEPT_DATA,wrt,noexe
.address   IR %extract(nameofs,namelen,ROUTINE);Intercept routine
.address   ROUTINE                                ;RTL routine
calls      #0,g^ROUTINE                          ;Fixup vector
.endm      INTERCEPT_TABLE

```


The INTERCEPT_TABLE macro is called with the name of the RTL routine that is to be intercepted. It automatically constructs a reference to both the user defined intercept routine and the system's routine. The code to call the RTL routine is never executed; it is simply included so that the fixup vector address can be determined. Invoking this simple macro is all that is needed to retrieve all of the necessary information. For example, when the above macro is invoked as follows:

```
INTERCEPT_TABLE    RTL_ROUTINE
```

the structure would be filled with the address of the intercept routine IR_RTL_ROUTINE as well as the address of the system's RTL_ROUTINE at run-time.

The fixup vector address is the key to this whole method since the contents of this location determine where all references to the RTL routine start executing.

Once the data structures are set up, all that is needed is to replace the addresses in the intercept vector with the addresses of replacement routines. Thereafter, all calls to the RTL routines will result in the replacement routines being invoked instead.

Since, using this technique, there is also access to the original RTL addresses, the RTL routines may be invoked freely from within the intercept routines.

The following code fragment is the entire code needed to implement the interception of RTL calls:

```

.psect      INTERCEPT_CODE,wrt,exe
.entry      InterceptInit,^m<r2,r3,r4>

movl  #NUM_OF_IRTN,r2      ; Number of routines to intercept
movab InterceptRoutines,r3 ; Address of intercept table

10$:                                ; FOR (each intercepted routine)
                                ; Calculate address of fix up
                                ; vector which is PC relative

addl3 #INTERCEPT_TABLE_SIZE,r3,r4
addl2 RTL_FIXUP_VECTOR(r3),r4

movl  r4,InAdr              ; Save address to be unprotected in
movl  r4,InAdr+4            ; system service argument block

clrl  -(sp)                 ; Set up
pushl #PRT$C_UW            ; call for
clrq  -(sp)                 ; system service
pushal InAdr               ; to unprotect
calls #5,g^SYSS$SETPRT     ; the fix up
blbc  r0,15$               ; vector area

movw  @RTL_ADDR(r3),-       ; Replace the reg save mask in our
@IR_ADDR(r3)               ; intercept routine with RTL and
movl  IR_ADDR(r3),(r4)      ; change the fixup vector in the
                                ; RTL to point to our routine

addl2 #INTERCEPT_TABLE_SIZE,r3; NEXT
sobgtr r2,10$

15$:
ret

```

Note that most of the code involves changing the memory protection on the fixup vectors, which requires special process privileges. Very little is needed to replace the actual addresses.

Case Study: FORTRAN WRITE Statements

At SimLab, all FORTRAN WRITE statements executed in real-time are intercepted and routed to an inter-process queue (pipe). The process on the receiving side of the queue runs at a much lower priority than the real-time process, and is allowed to handle these FORTRAN WRITE requests on a time-available basis. The real-time simulation is allowed to continue at a high priority without having to wait for the actual WRITE request to complete.

As an additional enhancement, this technique allows redirection of output from the simulation model while it is executing. Output can be routed to various printers, the terminal screen, or even a disk file (figure 5). A Motif X-Windows interface has also been developed which allows these FORTRAN WRITE statements to be routed to it. All of this is completely transparent to the real-time simulation process itself.

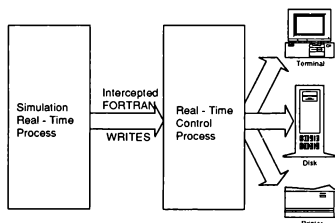


Figure 5. FORTRAN WRITES Data Flow

Intercepting the FORTRAN WRITE routines turned out to be more difficult than originally thought. This is primarily because in FORTRAN, the WRITE statement has many different parameters and forms. Consequently, the compiler generates many RTL calls for each WRITE statement instead of only one call, which is the usual case.

Even though many VAX/VMS calls are generated for each WRITE statement, they all follow a predictable format and sequence. The

first call is to a FOR\$WRITE routine, followed by any number of FOR\$IO calls (determined by the number of variables specified in the WRITE *iolist*) and ending with a FOR\$IO_END call.

A global variable is used to determine whether the normal system code should be invoked. When the application is performing a memory WRITE, for example, the call should not be altered. The following diagrams (figures 6-8) illustrate the program logic employed to accomplish this.

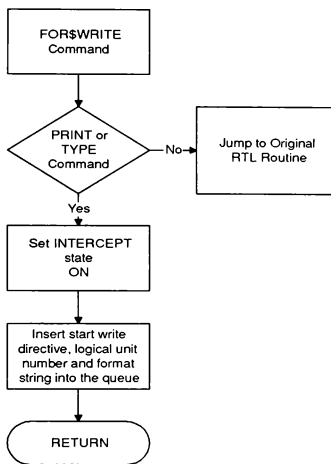


Figure 6. FOR\$WRITE Logic Flow

A call to the FOR\$WRITE routine specifies the FORTRAN unit number and format string. If the parameters to this routine do not specify a FORTRAN TYPE or FORTRAN PRINT command, then the original RTL is invoked. Otherwise the INTERCEPT flag is set and the parameters are passed through the queue.

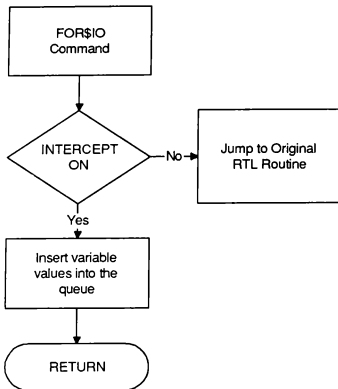


Figure 7. FOR\$IO Logic Flow

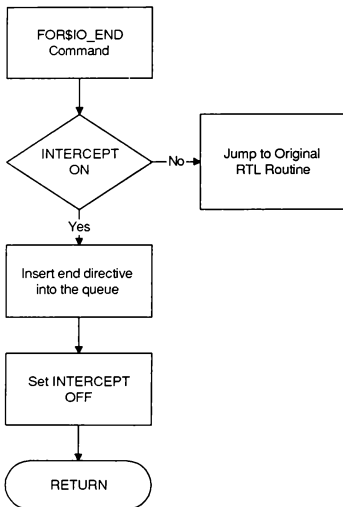


Figure 8. FOR\$IO_END Logic Flow

For each variable, a call to a FOR\$IO routine specifies the variable's data type (integer, floating, string, etc.), and either the variable's value or address. The actual values of the

variables are passed to the inter-process queue; otherwise, the values of the variables may change by the time the WRITE requests are processed.

The FOR\$IO_END routine signifies that all parameters and variables for the current WRITE have been specified. This information is passed down the queue and the INTERCEPT flag is reset for the next WRITE statement.

Measurements and Evaluation

Various benchmark tests were performed to quantify the amount of time saved using the RTL intercept technique, outlined herein. Multiple runs were performed, with and without RTL intercept, to make certain that the observed results are statistically significant.

A program was written which exercised each of the following 5 FORTRAN WRITE statements:

```

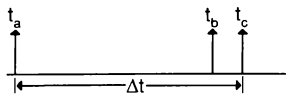
1 WRITE (6,*) I
2 WRITE (6,('INT = ',I)) I
3 WRITE (6,*) STR
4 WRITE (6,('STR = ',A40)) STR
5 WRITE (6,100) I1,R1,I2,R2,STR
  
```

where I, I1, and I2 are integer variables, R1 and R2 are floating point variables and STR is a character array variable.

Two test programs were built: one program used the normal RTL routines, the other was functionally identical but contained the replacement RTL routines. Output was directed to a disk file and to the terminal. Both programs were locked in memory and executed at the same real-time priority.

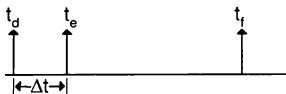
Information was collected to measure the elapsed time required to perform 1000 iterations of each of the test WRITE statements. The elapsed time measures the total amount of time from the start of the invocation of the RTL call until the real-time task resumes execution.

The time line in figure 9 shows that the elapsed time (Δt) for a normal RTL is $t_c - t_a$. The elapsed time for an intercept RTL, by comparison, is only $t_e - t_d$ (figure 10). Since the elapsed time is the time needed for the real-time process, the smaller the better.



t_a : Real Time Task issues I/O RTL call
 t_b : I/O completes
 t_c : Real-Time Task resumes execution

Figure 9. Normal RTL Time Line



t_d : Real Time Task issues I/O RTL call
 t_e : Real-Time Task resumes execution
 t_f : I/O completes

Figure 10. Intercept RTL Time Line

Table 1 contains the average *elapsed* times, in milliseconds (ms), observed for each of the five separate FORTRAN WRITE tests. The

normal disk write and normal terminal write values are the times observed using the operating systems' RTLs.

Test No.	Intercept RTL ($t_c - t_d$) (ms)	Normal Disk ($t_c - t_a$) (ms)	Normal Terminal ($t_c - t_a$) (ms)
1	00.004	00.173	12.204
2	00.004	00.457	30.678
3	00.024	00.267	16.803
4	00.026	00.442	32.053
5	00.040	00.985	76.599

Table 1. Average Elapsed Time for RTLs

Table 1 presents the order of magnitude differences observed between the intercept RTLs and both the normal disk and normal terminal RTLs. Figure 11 displays the same information as table 1 on a logarithmic scale, to more clearly illustrate these differences.

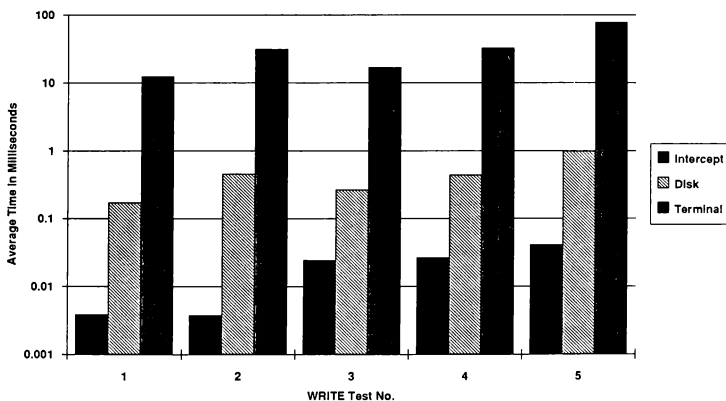


Figure 11. Average Elapsed Times for RTLs

As can be clearly seen, the differences for all the measured times are quite significant (over 1000%). As an example consider a simulation that has a 20 millisecond frame time. The time needed to queue a WRITE of one integer using the intercept method is 0.004 ms, while the normal method takes 0.173 ms to perform the WRITE (see Table 1 - Test 1). Logging 100 variables per frame (cycle) takes 0.4 milliseconds ($0.004 \times 100 = 0.4$) or only 2% of the total frame time when using the intercept RTL method. The normal method, by comparison, would take over 17 milliseconds ($0.173 \times 100 = 17.3$), nearly as great as the entire frame time.

The measured times had zero deviation for the intercept method, whereas over a 10% deviation was observed for the normal system routines. This is because the real-time process is entirely CPU bound during its WRITE routine. Knowing that there is no variance adds *determinism* to the real-time process.

In essence, most of the FORTRAN synchronous WRITES have been converted to asynchronous WRITES by using the RTL intercept technique. This results in dramatic improvements in real-time performance when FORTRAN WRITES are required.

Advantages and Disadvantages

Besides the obvious benefits of enhancing real-time performance, another advantage of this method is that control of the intercepted routines is dynamic; execution can be transferred and modified at run time, different paths can be executed depending upon the states of control structures, or the original routine can be executed, if desired. The fact that the real-time process is now entirely CPU bound adds the benefit of determinism to the time-critical tasks.

Additional benefits to the applications programmer are that (1) the programmer is not burdened with remembering special purpose routine names to handle common tasks such as real-time I/O, and (2) concerns about the priority and timing of these routines become less of an issue.

Intercepting RTL calls is not without its disadvantages. The programming tasks on the systems side become more complicated. Special privileges are usually required since the jump tables are normally located in a protected

memory space and great care is needed to synchronize access to these tables, especially in a multi-processor environment.

Conclusion

Additional complexities notwithstanding, the advantages gained by intercepting RTL calls are numerous and dramatic. In addition to increased programming flexibility, the required execution time of the real-time process is substantially decreased. This technique can also be utilized in non-real-time applications, as well as on other computational platforms.

¹Angermeyer, John, Jaeger, Kevin (1986) MS-DOS Developers Guide, page 96. Howard W. Sams & Co. Indianapolis, IN. USA

²Digital Equipment Corporation (1991) Run Time Library Introduction. pages 1-19, Digital Press, USA.

³Goldenberg, Ruth E., Lawrence, Kenah J. (1991) VAX/VMS Internals v 5.3, pages 763-767. Digital Press, USA.

AN EVALUATION OF SOFTWARE TOOLS FOR THE DESIGN AND DEVELOPMENT OF COCKPIT DISPLAYS

Thomas D. Ellis, Jr.
Unisys Corporation
NASA Langley Research Center
Hampton, Va. 23681

ABSTRACT

The use of all glass cockpits at the NASA Langley Research Center (LaRC) simulation facility has changed the means of design, development, and maintenance of instrument displays. The human-machine interface has evolved from a physical hardware device to a software generated electronic display system. This has subsequently caused an increased workload at the facility. As computer processing power increases and the glass cockpit becomes predominant in facilities, software tools used in the design and development of cockpit displays are becoming both feasible and necessary for a more productive simulation environment.

This paper defines LaRC requirements of a display software development tool and compares two available applications against these requirements. As a part of the software engineering process, these tools reduce development time, provide a common platform for display development, and produce exceptional real-time results.

INTRODUCTION

LaRC Environment

The Advanced Real-Time Simulation System (ARTSS)¹ is comprised of multiple cockpits, operation consoles, out-the-window image generators, and cockpit display generators connected to a high speed fiber optic network. Mathematical model computations supporting these resources are performed by CONVEX supercomputers.

Two different computer systems provide display generation for the simulator cockpits. Terabit Eagle 1000 computers provide calligraphic, raster, and mixing capability on the majority of the cockpits. Silicon Graphics, Inc. (SGI) computer systems provide raster instrument displays for an advanced transport cockpit and target images for air-to-air combat simulation in a 40' full dome simulation facility. Both provide extensive graphics libraries for designing and animating instrument displays in a lower level computer language and have software development tools, hereafter called tools, available for the design of displays. The results presented in this paper are based on the analysis of TOOL1, a development package for Terabit computers, and TOOL2, a development package for SGI computers.

Display Development Environment

Display development begins at LaRC with a research requirement. From this, researchers design display prototypes on paper or their own computers. Once the display requirement is complete, specifications are passed to the simulation programmer who develops and maintains the display on the real-time graphics computer. This is an iterative cycle demanding significant communication during the requirements, design, and development processes. Figure 1 reveals the major processes in display development.

Tools applied to this environment produce major benefits. By offering a graphical mouse-driven development interface, these tools provide the researcher more power to participate in development. It also establishes a common

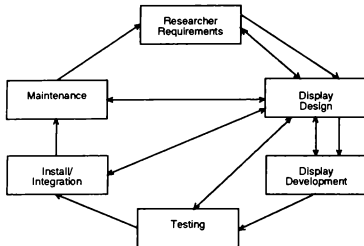


Figure 1 Display Development Cycle

working environment for researcher and programmer by eliminating multiple implementations of a display. Previous studies show development time for tools is reduced compared to traditional programming techniques.²

With these possible benefits, LaRC must determine the feasibility of tools on the simulation facility with respect to requirements.

LaRC REQUIREMENTS OF A SOFTWARE TOOL

Display Classifications

Cockpit displays can be broadly grouped into three increasingly complex categories; non-interactive, decision support, and prototyping. Non-interactive displays are read-only displays providing information based on model parameters passed from an external simulation. Decision support displays have read and write capabilities, allowing direct pilot interaction and control via touch-screen. Prototyping displays are a type of decision support display that perform virtual mock-ups of a complete cockpit panel.

Software Tool Requirements

If a tool is going to be used at LaRC, it must meet research, user, and facility requirements.

Research requirements for displays have evolved from non-interactive to decision support displays.

A tool must be capable of creating both types of displays.

User requirements arise from the specific needs of researchers and simulation programmers. A tool must allow graphical, mouse driven display development. This environment must also facilitate the creation of display dynamics.

Facility requirements are based on current system configurations and performance considerations. The tool must allow C Graphics Library (GL) source code generation on an SGI workstation, have acceptable real-time performance of generated displays, and provide a seamless communication environment with ARTSS.

To analyze how tools meet our requirements, TOOL1 and TOOL2 are thoroughly tested by developing a detailed primary flight display with each tool. From this development, comparisons are made based on the functional capability, real-time performance, and communication capability.

TOOL EVALUATION

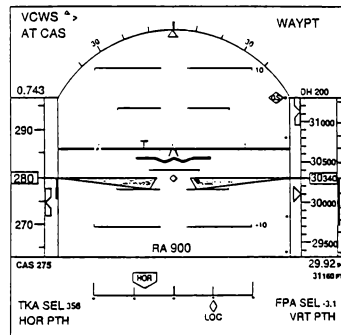


Figure 2 Primary Flight Display

Figure 2 illustrates the display developed using both tools. It includes an attitude director indicator, perspective runway, altitude tape and mode indicators, vertical acceleration indicator, airspeed tape and mode indicators, glideslope

Table 1 Functional Tool Capabilities

		TOOL1	TOOL2
DRAWING PRIMITIVES	line, rectangle, circle, arc, polygon, text	✓	✓
	chord, pie, tape, linear/circular labels, tick mark, primitive, viewport	✓	
	spline curve		✓
DRAWING OPERATIONS	copy, angular replicate, mirror, undo, delete, move, rotate, scale	✓	✓
	align	✓	
	round corners, push/pop, swap		✓
INPUT OBJECTS	button, rotor, switch, knob, potentiometer, input field, key, locator, menu		✓
OUTPUT OBJECTS	light, barchart, dial, scale, tape, CRT, cursor, output field, ADI, PPI, plot		✓
	User-defined output objects	✓	

scale, localizer scale, horizontal error scale, and all necessary mode and warning text strings.

General Overview of Tools

TOOL1 was originally designed to create and animate non-interactive displays for Terabit computers. It has recently developed the capability to develop decision support displays and generate GL compatible C code for SGI workstations. TOOL1 is contained within a single executable process and includes 73 point-and-click menu options for display creation and manipulation.

TOOL2 was originally designed to create and animate prototyping displays for SGI computers. It has recently developed a C source code generation capability. TOOL2 is contained within four primary executable processes; object creation, data integration, logic manipulation, and run-time environment. There are 86 point-and-click menu options in object creation and data integration.

Functional Capability

Both tools allowed adequate development of the evaluation display with the limitation that neither tool facilitated the perspective runway. (Note: TOOL2 has a recently developed 3D product that is available as a separate process.) After adequate training time, development took 20-25 hours for both tools.

Table 1 compares TOOL1 and TOOL2 based on available drawing primitives, drawing operations, input objects, and output objects. TOOL1 and TOOL2 provide display dynamics capability but differ in their approach. TOOL1 contains more drawing primitives than TOOL2 but no pre-set objects. Instead, all object dynamics are user-defined and customized. A logic script window based on C language syntax can be selected to describe an objects dynamic behavior.

TOOL2 facilitates dynamics in two ways. First, property sheets associated with pre-set objects (tapes, dials, plots) define the dynamic behavior of an object, allowing rapid development of

common objects without requiring knowledge of C language syntax. Second, for overall detailed display control a separate executable process controls dynamics through a finite state machine (FSM). The FSM is a logic editor where the user controls display behavior based on system input. This is ideal for prototyping displays where frequent user input such as button selection causes on-screen changes.

To reveal these different approaches to display dynamics, three small display development tasks were defined and developed on both tools. Two of these tasks had detailed display dynamics. The third task involved minimal display dynamics. Each task was performed five times to provide acceptable results. For the two tasks containing detailed display dynamics, TOOL1 performed 10-15% faster than TOOL2. For the third task, TOOL2 performed 15% faster than TOOL1.

The delay for TOOL2 in the first two tasks is due to swapping to the FSM process for detailed logic development. The TOOL1 logic script window allows direct entry of C code syntax, providing much flexibility in display dynamics. However, the presence of pre-set objects in TOOL2 helps to decrease development time as revealed in the third task.

Real-time Performance

Both tools rely on source code generation to produce acceptable real-time refresh rates. TOOL1 and TOOL2 provide options to generate Graphics Library (GL) C source code on SGI workstations. A SGI 4D/310 VGX is chosen as the common platform to compare the performance of the developed display for both tools. The performance measurement is based on the following; a monitor refresh-rate of 60 Hz, canned data driving the objects, and no real-time communication. Two cases are considered for timing, the display with vector fonts and with raster fonts. Results are summarized as follows:

	Raster Fonts	Vector Fonts
TOOL2	60Hz	30Hz
TOOL1	20Hz	20Hz

The TOOL2 generated primary flight display

performed better than TOOL1 in both cases.

TOOL2 took advantage of SGI optimization techniques to improve its speed by utilizing bitplane partitioning. Non-moving parts of the display are written into static underlays and overlays to allow only the moving part of the display to be rendered each frame. Tape and ADI objects are automatically clipped outside of the viewport range to avoid useless rendering. Negatively, TOOL2 generated C code required 60% more function calls than TOOL1 to draw the same display. Many of these functions are pre-set TOOL2 libraries with no provided source code. This makes debugging and code modifications difficult and in some cases impossible. Documentation of generated source code is thorough.

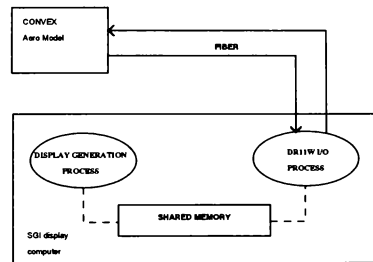


Figure 3 Current LaRC I/O Environment

TOOL1 uses fewer functions to generate the display than TOOL2, and all source code is available for debugging. However, TOOL1 does not take advantage of key optimization techniques used by TOOL2 and is greatly slowed by graphics look-up tables called in the program. Both tools exhibited performance degradation due to input device reads during each frame.

Communication Capability

ARTSS, the LaRC simulation facility, provides a serial fiber optic highway to transfer mathematical model variables calculated on the CONVEX supercomputer to each site configured on the network, and to receive input from the sites. For display generation, each SGI interfaces to the highway through a DR11W type interface. Figure 3 illustrates that two processes run in tandem to properly allow displays to run at real-time. The DR11W process reads data packets from the highway and places it into shared memory accessible by the display process.

Requirements such as this place tools at a disadvantage to support unique communication configurations. Both TOOL1 and TOOL2 require strict adherence to generated data structures, making turn-key solutions from tools difficult. Developers of tools are proactive about this situation and are beginning to develop communications support for developed displays. TOOL2 has an initial release of a communications library to allow external simulations to communicate with TOOL2 display processes. Illustrated in Figure 4, it is based on modifying the external simulation to read and write via TCP/IP to TOOL2 processes across a network.

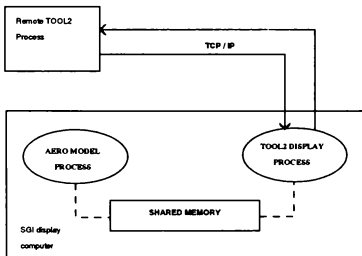


Figure 4 TOOL2 communication set-up

Although modifying the external simulation is undesirable for LaRC, generated displays from TOOL1 and TOOL2 can be used with an I/O process developed in-house to allow communication with ARTSS.

CONCLUSION

TOOL1 and TOOL2 fulfill many of LaRC's requirements of a tool with the notable exception of communication capability with ARTSS. Differences have been revealed in this analysis in display dynamic capabilities and source code generation. It is appropriate to discuss an ideal tool for LaRC to highlight these areas and how a tool best meets our requirements.

Functionally, the tool should contain all drawing primitives and operations similar to those listed in Table 1. In addition, pre-set objects like those listed for TOOL2 in Table 1 should be available to help decrease development time. As with TOOL1, logic scripts should optionally be available to tie directly to objects. This combines the efficiency of TOOL2 pre-set objects with TOOL1 custom object capability. An additional optional process should be available with FSM capability. This provides high level program control of complex decision support and prototyping displays.

For source code generation, the tool should have the capacity to produce C source code with GL functionality. All necessary source code for pre-set library functions should be provided for proper debugging. Optimization techniques involving bitplane partitioning and graphics cropping, as provided by TOOL2, must be performed.

For communications purposes, in addition to generating a display, the tool should generate an executable process that reads data from a DR11W interface and writes to shared memory for use by the developed display. This avoids modification requirements to external simulation or display code and the need to hand code an I/O process. This is the area where TOOL2 and TOOL1 did not meet LaRC's requirements.

This specification is a combination of the best of

both tools and LaRC's requirements. As of publication of this paper, both tools are continuously evolving, and to an extent, this paper provides only a snapshot of current capabilities.

REFERENCES

1. Crawford, D., Cleveland, J., & Staib, R. (1988 August). The Langley Advanced Real-Time Simulation (ARTS) System. AIAA paper 88-4595, Atlanta, GA
2. Nordwall, B. A340 Electronic Displays Created With Automated Software System. Aviation Week and Space Technology, February 4, 1991

REUSABLE CODE FOR HELICOPTER SIMULATION

Jo Ann Pickerine
Loral Defense Systems – Akron
Akron, Ohio 44315

Kuo-Chi Lin
Institute for Simulation and Training
Orlando, FL 32826

Curtis Lisle
Institute for Simulation and Training
Orlando, FL 32826

G. Guruprasad
Institute for Simulation and Training
Orlando, FL 32826

ABSTRACT

Truly reusable software is software that is defined to be reused without change. A method for producing reusable simulation software, which results in greatly reduced development and maintenance costs, has been developed by the Akron division of Loral Defense Systems and is now being tested in conjunction with the Institute for Simulation and Training, and the Naval Training and Simulation Center, both in Orlando. The Ada programming language is used for this effort.

The **reusability, without change**, of the software is achieved by a five-step method:

1. Make the system to be simulated **object-oriented**. That is, partition the system into a collection of objects to be simulated.
2. For each object, create a package of **single-function, untailored routines** which simulate that object's behavior. Each routine must perform one and only one main function, and perform that function in the most general manner possible. The packages of routines are then stored in the subsystem of a library for reusable code, according to the type of object they simulate. Any platform specifics for a particular simulation must be packaged separately and stored in their own library. These can also be reused. When the object's simulation is broken down into the small single-function routines, the routines can be distributed as desired over any platform. This, and the fact that platform specifics are kept separate, make the routines **platform Independent** also.
3. For each of the object's routines, create an **interface module**. This module must have a broad interface with the system, but a narrow interface with its routine.

4. Give each object's simulation a **generic package** that lists, and assigns generic names to, the modules used in the simulation.

5. Make all the interfaces and their routines **units Independent**, and make all the routines **data structure Independent**.

As each new simulation is being developed, the new system is partitioned into objects, and the corresponding subsystems of the reusable library are inspected for the needed object simulations. Those object simulations that are present will be reused without change. Those that are not will be developed, according to the above dictates, and added to the appropriate subsystems of the reusable code library. However, if an object simulation is present but one (or more) of its routines does not perform its function using a method found necessary for this particular application, another routine can be coded using the desired method. But here also, the function must be performed in the most general manner possible for the method employed. The new routine (or routines) is put into its own package and the present object simulation is used, but with the new routine being executed instead of the old. This is accomplished by means of a generic package.

The generic package is used to list the routines that will simulate the object in this particular application, and the package replaces the old routine in the list with the new routine. The replacement is facilitated by the interface modules of the two routines. This use of the generic package allows for the development, storage, and maintenance of unique object simulation routines only, and for the "mixing and matching" of these routines to produce any desired object simulation.

Here again, the routines' interface modules facilitate the "mixing and matching."

INTRODUCTION

Simulation software reusability is of particular advantage at this time, when a great deal of new software is being developed due to the military's mandate for Ada in its embedded computers. Also, software reusability is now a feasible concept for real-time simulations, due to the developments in processor memory size and execution speed. No longer does simulation software have to be tailored to fit the available memory size or to execute completely in a given amount of time, by eliminating portions that have little or no effect on the particular simulation being developed.

This means that much of the software developed for one simulation can be reused for similar simulations. For example, all bodies in flight are subject to the same physical laws of motion which are expressed in classic mathematical equations. In the simulation of an aircraft, if the portion of the software simulating the motion of the aircraft's body is kept separate from the software simulating the other parts of the aircraft, and if this software reflects the classic equations, it can be reused to simulate the motion of the body of any kind of aircraft. If the software simulating the aircraft's rotor is handled the same way, it can be reused to simulate the rotor on any kind of aircraft having the same type of rotor. After extending this principle to all parts of the aircraft simulation, it becomes clear that the software not only must be in the Ada language because of the government mandate, but also that it must be object oriented and untailored. This is the foundation for software reusability. Further, if the software simulating an object is made units independent, data structure independent, and platform independent, reusability of the object's simulation as a whole is maximized. Breaking an object's simulation software into single-function routines, and giving the routines each an interface module with the result that all routines in the object's simulation interface uniformly with the system, enable the reusability of the parts of an object's simulation.

THEORY

Standard object-oriented programming has the data structure holding an object's physical characteristics, data values which are either constant or repeatedly calculated, packaged with the routines that operate on these characteristics to simulate the object. This method separates the characteristics (Figure 1A) from their routines and passes

them to the routines through arguments (Figure 1B and 1C). The package of characteristics is now the object, and the package of routines the object's simulation. This allows the routines to be reused to simulate any of a class of unique but similar objects, those having the same data structure but different values for the constant characteristics, while maintaining only one copy of the routines package. The unique object packages can also be stored and reused.

NOTE: All further references to an object will imply a class of similar objects.

Platform independence is achieved through the breakdown of all objects' simulation software into single function routines for ease of distribution over any platform, and the separation of platform specifics from the routines. Platform specifics are packaged separately and can be reused. The routines must also perform their functions in a most general manner to help maximize reusability. Again referring to the calculations for the motion of a body in flight, this method has one routine calculating only the linear velocity components, another routine calculating only the linear acceleration components, etc., and each calculation is done with classic equations. No terms in any of a routine's equations can be omitted, nor can the equations be otherwise tailored to a specific body's motion. If either of these occur, the routine can be reused without change only for the simulation of the motion of that particular body, rather than for a class of similar bodies.

Each routine in an object's simulation is given an interface module (Figure 1Ea). When execution of a routine is needed, a call is made to its interface module, which in turn calls the routine. The package of single-function routines becomes a package of interface modules with a routine tied to each of them. Because of the different methods for simulating an object, there could be more than one package of modules and routines for an object. The modules in all packages, with simulation routines for the same object, must interface with the simulation system in exactly the same general manner. That is, all of these modules accept the same set of arguments. These arguments are large portions of the system data structures, portions large enough to contain all possible data needed for all the object's routines to perform their functions by any method chosen.

Having the modules interface with the system in this manner, and having their routines singular in function, makes possible the interchange of two routines that perform the same function but

A. Object package for Body

```

.
.
.
type Body_type is
  record
    Linear_Acceleration : Lin_Accel_type;
    Linear_Velocity : Lin_Vel_type;
    Delta_Time : Time_Inc_type;
    .
    .
    .
  end record;
.
.
.

```

B. Collective object package for Helicopter (optional)

```

.
.
.
type Copter_type is
  record
    Body : Body_type;
    .
    .
    .
  end record;
.
.
.
Helicopter : Copter_type;

```

C. Calling program

```

.
.
.
Calc_Linear_Velocity (Helicopter.Body);
.
.
.

```

D. Generic package

```

.
.
.
with Procedure Calc_Linear_Velocity (Body : In Out Body_type) is
Flight_Calcs.Calculate_Linear_Velocity;
.
.
.

```

E. Object simulation package – Flight_Calcs

```

Interface Module (a)
Procedure Calculate_Linear_Velocity (Body : In Out Body_type) is
.
.
.
  Integrate_Linear_Acceleration (Linear_Acceleration,
                                Linear_Velocity,
                                Delta_Time);
.
.
.
end procedure;

Routine (b)
Procedure Integrate_Linear_Acceleration (Lin_Accel : in Lin_Accel_type;
                                         Lin_Vel : Out Lin_Vel_type;
                                         Delta_T : in Time_Inc_type) is
.
.
.
end procedure;

```

Figure 1. Method For Developing Reusable Code

perform it by different methods. For example, two routines that both integrate linear acceleration to produce linear velocity, one by the rectangular method, the other by the trapezoidal method, are interchangeable. If, in order to reuse an object's simulation in a new development effort, a routine must be removed from the simulation and another inserted, the interface module facilitates the interchange and allows the reuse without change of the rest of the routines. This is because it is actually the interface modules that are interchanged and the modules preserve the argument list.

To further facilitate the interchange, a generic package (Figure 1D) is used to list the modules for the routines to be used in an object's simulation in a new development effort. In listing the modules, the generic package associates a generic name with each of them, and calls to these modules are made using the generic names, which never change. Along with unchanging argument lists, this eliminates any need for modifications in the calling programs because of the interchange. Modifications are confined strictly to the generic package and involve only the interface module name and the package name for the replaced routine. However, modules whose routines perform the same function can be given the same names, since they will reside in packages having dissimilar names which will differentiate between the two. Now even the modifications to the generic package are minimized. The interchange becomes nothing more than the interchange, in the generic package, of the names of the two packages in which the modules and their routines reside. Moreover, the introduction of the interface module and the generic package not only facilitate the interchange of two routines, but make possible the "mixing and matching" of routines to put together any desired object simulation.

While a routine's module interfaces with the system in a general manner, it interfaces with its routine in a very detailed manner dictated strictly by the needs of the routine (Figure 1Eb). The module passes to the routine only those single pieces of data needed for the performance of the routine's function. Having the routines accept only single pieces of data makes the routines data structure independent. They are oblivious to whatever kind of structure holds the data.

If newly coded routines need to store pieces of data in the object's data structure and the structure is not set up for it, the structure can be expanded to accommodate the new data. The expanded structure can then be used along with the

new routines and without affecting the already written routines. The interface modules of the already-written routines accept whole structures regardless of additions. Their only concern with the pieces of data is that the pieces needed by their routines are still in the structure.

The objects, routines, interface modules, generic packages, etc. are created independent of units. This is accomplished by using actual names of desired units to define default units types in a separate package (Figure 2A). Various necessary unit conversion factors and functions are also put into the package. Generic types are then defined from the default units types and generic names are given to the conversion factors and functions. Both the generic factor and function names and the generic types are used throughout the various packages in the reusable library (Figure 2C). To convert a simulation from one set of units to another, from degrees, feet, miles, etc., to radians, meters, kilometers, etc., one need only

A. Units package A

```
.
.
.
type Degrees is Float;
type Angle_type is Degrees;
.
Convert_to_Radians : Float := Pi/180;
.
.
```

B. Units package B

```
.
.
.
type Radians is Float;
type Angle_type is Radians;
.
Convert_to_Radians : Float := 1.0;
.
.
```

C. Routine

```
.
.
.
Temp_Var : Float;
Heading : Angle_type;
.
.
Temp_Var := Sin (Convert_to_Radians * Heading);
.
.
```

Figure 2. Units Independence

remove the package containing the default units types formed from the first set of units, and replace it with the package containing default units type formed from the desired set of units (Figure 2B). No modifications need to be made to any other part of the simulation.

TESTING

This method for producing reusable software is being tested through its application to the existing simulations of two helicopters, which will be referred to as helicopter A and helicopter B. The actual names of the helicopters are irrelevant, since the application of the method is independent of the choice of helicopters. The two simulations were originally programmed in the FORTRAN language and are being reprogrammed in Ada, according to the method developed by Loral. Helicopter A is being reprogrammed first with all routines possible being reused in the reprogramming of helicopter B. In the simulation of helicopter A, the main rotor force and moment calculations are performed using the disc rotor method (Figure 3C and 3D). In the simulation of helicopter B, the same main rotor calculations are performed using the blade element method (Figure 4A and 4B). All other functions are common to the two simulations. The packages of routines that perform these common functions, after being reprogrammed first for helicopter A, are being reused without change in the development of the simulation of helicopter B in Ada. The only routines that need reprogramming for helicopter B are the routines performing the main rotor's force and moment calculations (see Figure 3).

However, there are differences in the types and physical characteristics of some of the components of the two helicopters, such as the engine and body. This results in the need to store different characteristic constants in the object packages for these components in the helicopter B simulation. Therefore new object packages, and any related data table packages for thrust, lift,

drag, etc., were created for these components (Figure 5).

Since so much of the code for the simulation of helicopter A could be reused without change, the development costs for the simulation of helicopter B were drastically reduced. Testing time for the reused code was also reduced to testing its interaction with the newly developed routines. Maintenance costs for shared enhancements, fixing discrepancies and such will also be greatly reduced. In short, the more the code can be shared without change, the greater the reduction in effort and cost.

SUMMARY

Reusable simulation code is code that can be reused without change. To obtain this degree of reusability:

1. The code needs to be **object-oriented**, with the object packaged separately from the simulation of the object.
2. The simulation of the object must consist of a package of **single-function routines** that are **untailored**. The singularity of function allows the routines to be **platform independent**.
3. Each routine must be given an **interface module** to ensure unchanging argument lists.
4. Each object simulation must be given a **generic package** that lists the modules used in the simulation of the object and associates generic names with them to ensure unchanging module names in the calling programs.
5. The routines and modules must be **units independent** and the routines must be **data structure independent**.

Producing reusable code, code that can be reused without change, results in reduced development and maintenance costs. The greater the percentage of reusable code incorporated into the development of a new simulation, the greater the immediate and long-term savings.

A. Collective object package for Helicopter

```

.
.
type Copter_type is
  record
    Body : Body_type;
    .
    .
  end record;
.
.
Helicopter : Copter_type;

```

B. Calling program

```

.
.
Calc_Rotor_Forces_and_Moments (Helicopter);
.

```

C. Generic package

```

.
.
with Procedure Calc_Rotor_Forces_and_Moments (Any_Copter : In Out Copter_type) is
  Disc_Rotor_Calcs.Calc_Rotor_Forces_and_Moments;
.
.

```

D. Object simulation package for Rotor A – Disc Rotor Method

Interface Module

```

Procedure Calc_Rotor_Forces_and_Moments (Any_Copter : In Out Copter_type) is
.
.
  Calc_Forces_and_Moments ( . . . );
.
.
end procedure;

```

Routine

```

Procedure Calc_Forces_and_Moments ( . . . ) is
.
.
end procedure;

```

Figure 3. Rotor Simulation For Helicopter A

A. Generic package

```

.
.
.
with Procedure Calc_Rotor_Forces_and_Moments (Any_Copter : In Out Copter_type) is
Blade_Element_Rotor_Calcs.Calc_Rotor_Forces_and_Moments;
.
.
.

```

B. Object simulation package for Rotor B – Blade Element Method

Interface Module

```

Procedure Calc_Rotor_Forces_and_Moments (Any_Copter : In Out Copter_type) is
.
.
.
Calc_Forces_and_Moments ( . . . );
.
.
.
end procedure;

```

Routine

```

Procedure Calc_Forces_and_Moments ( . . . ) is
.
.
.
end procedure;

```

Figure 4. Changes For Rotor Simulation For Helicopter B

A. Object package for Body of Helicopter A

```

.
.
.
type Body_type is
record
Linear_Acceleration : Lin_Accel_type;
Linear_Velocity : Lin_Vel_type;
Delta_Time : Time_Inc_type;
Weight : Constant := 12000.0;
.
.
end record;
.
.
.

```

B. Object package for Body of Helicopter B

```

.
.
.
type Body_type is
record
Linear_Acceleration : Lin_Accel_type;
Linear_Velocity : Lin_Vel_type;
Delta_Time : Time_Inc_type;
Weight : Constant := 17000.0;
.
.
end record;
.
.
.

```

Figure 5. Change In Body Object

REFERENCES

Prouty, R. (1990). *Helicopter Performance, Stability, and Control*. Robert E. Krieger Publishing Company, Malabar, FL.

Gessow and Myers (1952). *Aerodynamics of the Helicopter*. The MacMillan Company, New York, NY.

Roskam, Jan (1982). *Airplane Flight Dynamics and Automatic Flight Controls Part I*. Roskam Aviation and Engineering Corporation. Ottawa, KS.

Howlett, J. J. (1981). *UH-60A Black Hawk Engineering Simulation Program*. NASA Contractor Report 166309. Ames Research Center, Moffett Field, CA.

American National Standards Institute, Inc. (1983). *Reference Manual for the Ada Programming Language*. ANSI/MIL-STD-1815A-1983.

Bergsneider, Carl (1992). *Reusable Helicopter Modeling For Simulation*. Simulators IX, Simulation Series Volume 24, Number 4. Society For Computer Simulation.

Pickering, Jo Ann (1993). *Reusable Code Structures For Simulation*. Simulators X, Simulation Series Volume 25, Number 4. Society For Computer Simulation.

BIOGRAPHIES

Jo Ann Pickering received her B.S. in Mathematics at Carnegie-Mellon University in 1983 and was employed by Loral Defense Systems – Akron shortly after.

She has worked on the F15-A Project in the software systems for the Modification Support, Graphics, Radar, and Data Transmission areas, and on the F15-E Project on the Multi-Task Multi-Processor Real-Time and Non-Real-Time Debug-

ger. She is currently a Scientific Systems Analyst for Loral in the Independent Research and Development Group of the Department of Simulator and Trainer Engineering.

Dr. Kuo-Chi Lin received his B.S. and M.S. degrees in Mechanical Engineering from the National Cheng-Kung University, Tainan, Taiwan in 1975 and 1979, respectively. His M.S. and Ph.D. degrees in Aerospace Engineering were received from the University of Michigan in 1986 and 1990, respectively.

He is a joint faculty member in the Department of Mechanical and Aerospace Engineering and The Institute for Simulation and Training at the University of Central Florida. His research interests include real-time simulation, flight simulation, inverse simulation, numerical integration algorithms, multibody dynamics, and robotics.

Curtis R. Lisle received his B.S. degree in Electrical Engineering from the Georgia Institute of Technology in 1986, and his M.S. degree in Computer Science from the University of Central Florida in 1990. He is currently pursuing his Ph.D. degree at the University of Central Florida.

He is a Visual Systems Scientist at the Institute for Simulation and Training at the University of Central Florida. His research interests include virtual reality, dynamic terrain data base, computer image generation, and solid geometry modeling.

G. Guruprasad received his B.E. degree in Mechanical Engineering from the University of Mysore, Mandya, India in 1986, and his M.S. degree in Mechanical Engineering from the University of Central Florida in 1992.

He is currently pursuing his Ph.D. degree and is a Research Assistant at the University of Central Florida.

FLIGHT UPDATE OF AERODYNAMIC MATH MODEL

Kendall W. Neville* and A. Thomas Stephens
Boeing Commercial Airplane Group
Renton, Washington

ABSTRACT

Increasingly stringent standards for simulator fidelity have led to a need for continually improving the aerodynamic mathematical models for flight simulations produced by Boeing Commercial Airplane Group (BCAG). An overview of the process used at BCAG for producing and validating the flight updated aerodynamic model is presented. Topics include flight-test planning, instrumentation requirements, aerodynamic coefficient extraction, and simulator-to-flight validation. Particular emphasis is given to the extraction method which is used to identify aerodynamic coefficient information for the entire model in both linear and nonlinear regions, including the effects of flow separation and aerodynamic hysteresis. An actual example is provided showing the flight updated basic lift coefficient compared to predicted data, and the resulting improvement of the simulator-to-flight comparison of a deceleration to full stall and the subsequent recovery.

NOMENCLATURE

b	Reference wing span, ft
\bar{c}	Reference wing chord, ft
C_{XAERO}	Body x-axis force coefficient
C_{YAERO}	Body y-axis force coefficient
C_{ZAERO}	Body z-axis force coefficient
$C_{\ell AERO}$	Body-axis rolling moment coefficient
$C_{m AERO}$	Body-axis pitching moment coefficient
$C_{n AERO}$	Body-axis yawing moment coefficient
C_{XSIM}	Simulator body x-axis force coefficient
C_{YSIM}	Simulator body y-axis force coefficient
C_{ZSIM}	Simulator body z-axis force coefficient
$C_{\ell SIM}$	Simulator body-axis rolling moment coefficient
$C_{m SIM}$	Simulator body-axis pitching moment coefficient
$C_{n SIM}$	Simulator body-axis yawing moment coefficient
F_{XT}	Body x-axis thrust force, lb
F_{YT}	Body y-axis thrust force, lb
F_{ZT}	Body z-axis thrust force, lb
$M_{\ell T}$	Body-axis thrust rolling moment, ft-lb
M_{mT}	Body-axis thrust pitching moment, ft-lb
M_{nT}	Body-axis thrust yawing moment, ft-lb

I_{xx}	Roll moment of inertia, slug-ft ²
I_{yy}	Pitch moment of inertia, slug-ft ²
I_{zz}	Yaw moment of inertia, slug-ft ²
I_{xz}	Product of inertia, slug-ft ²
MAC	Mean aerodynamic chord, ft
n_x	Body x-axis load factor
n_y	Body y-axis load factor
n_z	Body z-axis load factor
p	Body axis roll rate, rad/sec
q	Body axis pitch rate, rad/sec
r	Body axis yaw rate, rad/sec
\bar{q}	Dynamic pressure, lb/ft ²
S	Reference wing area, ft ²
W	Airplane gross weight, lb
α	Airplane angle of attack, deg
ΔC_X	Difference coefficient, body x-axis
ΔC_Y	Difference coefficient, body y-axis
ΔC_Z	Difference coefficient, body z-axis
ΔC_{ℓ}	Difference coefficient, body-axis rolling moment
ΔC_m	Difference coefficient, body-axis pitching moment
ΔC_n	Difference coefficient, body-axis yawing moment
ΔC_L	Lift coefficient difference between required and computed value, stability axes
$C_{L-SIM EXT}$	Flight-test-extracted total lift coefficient
$C_{L-BASIC EXT}$	Flight-test-extracted basic lift coefficient

INTRODUCTION

Modern standards for commercial transport aircraft flight simulators require a close match with airplane characteristics throughout the flight envelope. Simulations developed at Boeing Commercial Airplane Group (BCAG) must satisfy the stringent requirements of the highest level full-flight training simulators and those of its own engineering simulators. The aerodynamic mathematical model provides the foundation of each simulation and is designed to satisfy the requirements of both of these applications.

* Member AIAA

Crew Training Simulator

The requirements for the content and fidelity of the aerodynamic model are determined from several sources. The basis for crew training simulator requirements is the "International Standards for the Qualification of Airplane Flight Simulators"¹, published by the Royal Aeronautical Society. This document was developed by a team of representatives from all facets of the simulator industry, including the major regulatory authorities, who met regularly over a period of two years from 1989 to 1991. It is based on the content of FAA Advisory Circular AC120-40B² and expanded to include specific requirements of European and Australian regulatory agencies. Since the International Standards draft document was completed in January, 1992, it has been endorsed by major regulatory authorities as a means to achieve training simulator approval. As of this writing, it is awaiting formal approval of the International Civil Aviation Organization (ICAO).

Another important resource for training simulator standards is the document, "Flight Simulator Design & Performance Data Requirements"³, published by the International Air Transport Association (IATA). This document provides guidance for the scope and content of data necessary to build simulators of adequate fidelity to meet flight crew training requirements.

Engineering Simulator

At BCAG, the engineering simulator has become an essential analysis tool which is used for a variety of applications. A few examples are:

- system design
- system certification
- accident/incident investigation
- handling qualities and crew workload evaluations
- failure analysis
- customer requests
- customer demonstrations

The engineering simulator must accurately represent a wide range of conditions within the flight envelope and be readily available when needed.

Flight Update

A key to providing the required simulator fidelity for all of these applications is the flight update to the aerodynamic mathematical model. The process at

BCAG uses measured flight data to determine the incremental coefficient values which must be applied to the existing simulation aerodynamic model to satisfy the equations of motion.

Figure 1 provides an overview of the process for acquiring flight-test data, data conditioning, coefficient extraction, and time history matching.

FLIGHT TEST REQUIREMENTS

At BCAG, a significant portion of each flight-test program for new and derivative airplanes is devoted to gathering data for the development and update of the simulator aerodynamic math model. During one recent derivative program some 36 flight hours were devoted to simulator-specific testing. In addition, data acquired from other testing, such as that for performance and handling qualities certification, are used for simulator update.

Simulator flight testing requires careful planning. Test conditions, maneuver types, and pilot technique must be well defined. Onboard instrumentation must be specified along with the requirements for measurement accuracy. All of this planning requires close coordination between aerodynamics engineering, flight-test support groups, and the test pilots. Boeing test pilots are highly skilled and often provide valuable insight and suggestions for how a test should be performed. Particularly demanding maneuvers may be practiced in the engineering simulator prior to the actual test flight.

Two basic types of tests are conducted during the simulator-specific flight testing. The first includes all the handling qualities and performance tests required by the International Standards document¹ and other validation tests required by customers and regulatory authorities. Selected data from these tests will ultimately appear in the Approval Test Guide which is provided by each training simulator operator to the appropriate regulatory authority. The second type of testing includes those maneuvers which are designed for use in extracting coefficient data for the simulator aerodynamic math model. Each of these tests must be carefully defined in terms of flight condition, airplane configuration, initial condition, and pilot control inputs.

Flight Test Instrumentation

Simulator-specific flight testing typically occurs shortly after FAA certification of new and derivative BCAG airplanes. This sequencing guarantees that no configuration changes will occur during simulation testing and provides access to an aircraft which has been well instrumented and calibrated in support of pre-certification and certification flight testing.

From the hundreds of available flight test measurements, approximately 300 are selected for each simulator-specific maneuver. These data are collected into a special database of flight test maneuvers resident with the aerodynamics engineering group (see Figure 1).

The parameters requested for the simulator-specific flight test maneuver database are required to support two major tasks involved with the aerodynamic model update: 1) simulator model validation and 2) aerodynamic coefficient extraction. The parameters required to perform these tasks are:

- angle of attack
- sideslip angle
- control surface positions
- Mach number
- pressure altitude
- airspeed
- configuration (e.g. flaps, gear, etc.)
- engine thrust
- Euler angles
- angular rates
- linear accelerations
- angular accelerations
- mass properties
- winds

In addition, a fundamental parameter required for all maneuvers is time. It is important to remove all time skews between parameters so that any given event will occur at the proper time in each parameter.

Each of the required parameters mentioned above may be obtained from instrumentation available on the flight test aircraft. Two sources of these measurements are: 1) production ship system instrumentation and 2) instrumentation added specifically for flight test data acquisition. The production instrumentation includes an Inertial Reference Unit (IRU), Air Data Computer (ADC), angle of attack vane, engine instrumentation (e.g. throttle lever angles, N1, N2 etc.) and configuration measurements. Additional flight test instrumentation includes control surface positions, sideslip pressure ports, static pressure from a fin trailing cone, and total pressure valid to high air-flow angles from a shielded pitot tube.

Data Accuracy Requirements

It is necessary for analysis engineers to be familiar with the characteristics of each flight test measurement. The extracted aerodynamic coefficients are only as good as the flight test measurements used to compute them.

The term "measurement accuracy" refers to the degree to which the flight test measurement represents the actual

value. Measurement accuracy may be affected by: instrument accuracy (how well the instrument output represents the actual signal), resolution (how well the digital signal represents the analog signal), filtering, calibrations, and assumptions or approximations made during data conditioning.

Measurement uncertainties adversely affect both the accuracy of aerodynamic coefficients derived from flight test data and the simulator-to-flight validation. Measurement inaccuracy consumes some portion of the validation tolerance. In addition, measurement uncertainties may increase the scatter or noise in extracted coefficient data from different maneuvers and flights.

Measurement accuracies have improved at BCAG with the use of ring laser gyros for angular rates and euler angles and digital synchros for control surface deflections. These measurements are currently sufficiently accurate for aerodynamic model development. Further improvements to math model fidelity could result from increased accuracy of the following measurements: static pressure where trailing cone data are invalid or unavailable, free-stream angle of attack and sideslip angle, wind, thrust and mass properties.

The aerodynamic model update process used at BCAG undergoes continuous improvement. This process is approaching the point where further significant model fidelity improvements will require a reduction of flight-test measurement uncertainties.

Atmospheric Conditions

It is not possible with current technology to accurately account for the effects of a turbulent wind field on the aircraft response. Therefore, an important requirement for each test is that the atmosphere be as free of turbulence as possible. However, there inevitably will be some turbulence and small gusts during flight testing.

Accurate methods for measuring airspeed must be used. The airspeed vector is resolved into longitudinal, vertical, and lateral components. These airspeed components are compared to the inertial rates along each axis to determine the wind components. However, these computed wind values are valid only for the point at which the airspeed is measured (typically well away from the wing and empennage surfaces), and generally do not properly account for the high-frequency components of the wind field.

Calm air is especially important for tests near the ground. A significant wind occurring near the ground is

usually accompanied by turbulence. As a rule, a mean wind speed of less than three knots is required for ground-effect testing. The preferred time of day for ground-effect testing is near dawn in a stable atmosphere where there is likely to be a temperature inversion near the ground (the air temperature increases with height above the ground). There have been occasions during flight testing for ground effects when the air was so still that the first test of the morning was performed in virtually calm air only to have the airplane fly through its own wake turbulence on the subsequent test. The ideal atmospheric condition for ground-effect testing would be to have just enough air motion to carry the wake turbulence away from the runway before the next approach, but light enough to cause no atmospheric turbulence. Such an ideal condition is rarely encountered, and is always short-lived.

AERODYNAMIC COEFFICIENT EXTRACTION

Methods for aerodynamic simulator data extraction are key to successful simulator update. An overview of the method used at BCAG is provided below. The method uses information measured on board the flight-test aircraft and compares the forces and moments required to satisfy the airplane equations of motion with those computed by the existing simulation. The simulation is initially based largely on wind tunnel data and is continually revised during the update process. The differences between the required and computed aerodynamic coefficients are used to update the simulator aerodynamic data base. The method can be used to extract aerodynamic coefficient information in the linear and nonlinear regions, including the effects of flow separation and aerodynamic hysteresis. These methods rely on powerful computer-based tools, but considerable interpretation and engineering judgement is required to properly distribute the coefficient increments to the individual components of the aerodynamic buildup equations.

The coefficient extraction method utilizes the following six-degree-of-freedom equations of motion in airplane body axes:

$$C_{X_{AERO}} + \frac{F_{XT}}{qS} = \frac{n_X W}{qS} \quad (1)$$

$$C_{Y_{AERO}} + \frac{F_{YT}}{qS} = \frac{n_Y W}{qS} \quad (2)$$

$$C_{Z_{AERO}} + \frac{F_{ZT}}{qS} = \frac{n_Z W}{qS} \quad (3)$$

$$C_{\ell_{AERO}} + \frac{M_{\ell T}}{qSb} = \frac{[\dot{p}I_{XX} - (\dot{r} + pq)I_{XZ} + (\ell_{ZZ} - I_{YY})qr]}{qSb} \quad (4)$$

$$C_{m_{AERO}} + \frac{M_{mT}}{qSc} = \frac{[\dot{q}I_{YY} - (\ell_{ZZ} - I_{XX})pr - (r^2 - p^2)I_{XZ}]}{qSc} \quad (5)$$

$$C_{n_{AERO}} + \frac{M_{nT}}{qSb} = \frac{[\dot{r}I_{ZZ} - (\dot{p} - qr)I_{XZ} + (I_{YY} - I_{XX})pq]}{qSb} \quad (6)$$

The left-hand sides of the above equations of motion are the applied forces and moments which are required to satisfy inertial equilibrium. It is assumed here that these applied forces and moments consist only of aerodynamic and thrust effects (i.e., the airplane is in the air). The right-hand side of each equation represents the airplane kinematic state and mass characteristics as determined from flight-test measurements.

It is assumed that the thrust effects (F_{XT} , F_{YT} , F_{ZT} , $M_{\ell T}$, M_{nT} , M_{mT}) can be determined with sufficient accuracy from onboard measurements and the simulator engine model. It is also assumed that the required total aerodynamic coefficient for each degree of freedom can be represented as the value computed by the existing simulation for the given aircraft state plus a difference coefficient, i.e.:

$$C_{X_{AERO}} = C_{X_{SIM}} + \Delta C_X \quad (7)$$

$$C_{Y_{AERO}} = C_{Y_{SIM}} + \Delta C_Y \quad (8)$$

$$C_{Z_{AERO}} = C_{Z_{SIM}} + \Delta C_Z \quad (9)$$

$$C_{\ell_{AERO}} = C_{\ell_{SIM}} + \Delta C_{\ell} \quad (10)$$

$$C_{m_{AERO}} = C_{m_{SIM}} + \Delta C_m \quad (11)$$

$$C_{n_{AERO}} = C_{n_{SIM}} + \Delta C_n \quad (12)$$

The difference coefficients may be shown in terms of the equations of motion by combining equation (1) - (6) with equations (7) - (12):

$$\Delta C_X = \frac{n_X W}{qS} - \frac{F_{XT}}{qS} \cdot C_{X_{SIM}} \quad (13)$$

$$\Delta C_Y = \frac{n_Y W}{qS} - \frac{F_{YT}}{qS} - C_{Y_{SIM}} \quad (14)$$

$$\Delta C_Z = \frac{n_Z W}{qS} - \frac{F_{ZT}}{qS} - C_{Z_{SIM}} \quad (15)$$

$$\Delta C_l = \frac{[\dot{p} I_{XX} - (r + pq) I_{XZ} + (I_{ZZ} - I_{YY})qr]}{qSb} \cdot \frac{M_{lT}}{qSb} - C_{l_{SIM}} \quad (16)$$

$$\Delta C_m = \frac{[\dot{q} I_{YY} - (I_{ZZ} - I_{YY})pr - (r^2 - p^2) I_{XZ}]}{qSc} \cdot \frac{M_{mT}}{qSc} - C_{m_{SIM}} \quad (17)$$

$$\Delta C_n = \frac{[\dot{r} I_{ZZ} - (\dot{p} - qr) I_{XZ} + (I_{YY} - I_{XX})pq]}{qSb} \cdot \frac{M_{nT}}{qSb} - C_{n_{SIM}} \quad (18)$$

The incremental moment coefficients are transferred to a reference center of gravity, usually 25% MAC.

The incremental coefficients may be computed as a function of time for a given flight-test condition. Each incremental coefficient represents the "correction" which must be added to the simulation to satisfy the equations of motion.

The next task, often the most formidable, is to determine which terms in the aerodynamic model buildup should be revised. This is a highly iterative task requiring considerable engineering judgement. The complete flight-test update is undertaken by a team of engineers who have extensive knowledge of the aerodynamic effects on aircraft performance, stability, and control. The team must carefully coordinate its efforts to produce a consistent model which satisfies the requirements for all types of maneuvers throughout the flight envelope.

Example Application

An important requirement for determining the proper distribution of the aerodynamic effects is the availability of flight-test maneuvers which can be used to isolate the desired aerodynamic effects. An example which demonstrates the effectiveness of this method for identifying aerodynamic coefficient information in a highly nonlinear region is the extraction of the basic airplane lift at high angles of attack. An airplane stall test may be used for this analysis. During this test, the airplane decelerates at a specified entry rate (usually about one knot per second) with the engines set at idle power. The airplane gradually pitches up until a minimum speed is reached, at an angle of attack which

typically is greater than that for maximum lift coefficient. The same test may be used to extract basic pitching moment and drag coefficients.

The total simulator aerodynamic lift, in airplane stability axes, consists of the sum of a number of individual contributions. For the purpose of this example, the lift buildup equation may be expressed as:

$$C_{L_{SIM}} = C_{L_{BASIC}} + C_{L_{STABILIZER}} + C_{L_{ELEVATOR}} + C_{L_{DYNAMIC}} + \text{SMALLER TERMS} \quad (19)$$

A difference coefficient of lift, ΔC_L , is determined by transforming to stability axes using the body-axis ΔC_X and ΔC_Z of equations (13) and (15) respectively:

$$\Delta C_L = -\Delta C_Z \cos \alpha + \Delta C_X \sin \alpha \quad (20)$$

The term ΔC_L represents the correction which must be applied to the lift equation to satisfy inertial equilibrium. The new or "extracted" total lift coefficient may be expressed as:

$$C_{L_{SIM \text{ EXT}}} = C_{L_{SIM}} + \Delta C_L \quad (21)$$

The left-hand side of equation (21), $C_{L_{SIM \text{ EXT}}}$, represents the total lift coefficient which is required to satisfy inertial equilibrium. Figure 2 shows the total lift coefficient which was extracted from a stall test which was performed during a recent Boeing commercial transport flight-test program. The extracted lift coefficient data represent the entire maneuver from the initial deceleration through the recovery. The extracted data are shown together with the updated simulator total lift coefficient for the same maneuver. Note the aerodynamic "hysteresis" effect showing the delay in flow reattachment during the recovery portion of the maneuver.

The method also may be applied to the extraction of an individual buildup term by using a maneuver for which that function is a primary contributor to the airplane response. The example stall test may be used to identify the basic lift function, $C_{L_{BASIC}}$. For this purpose, the required difference coefficient is applied in the following manner to determine a new "extracted" basic lift coefficient:

$$C_{L_{BASIC \text{ EXT}}} = C_{L_{BASIC}} + \Delta C_L \quad (22)$$

It is assumed that all the individual contributions to lift except for $C_{L_{BASIC}}$ are known from previous analysis or are relatively small. (Actually, the process is highly iterative, and as each term is modified, the other terms must be re-evaluated until all are well defined.) The left-hand side of equation (22) represents the new basic lift

coefficient. Figure 3 shows the pre-flight predicted basic lift coefficient compared to data extracted from the example stall test. After applying the correction to the basic lift coefficient, the function CL_{BASIC} is revised as shown in Figure 4.

VALIDATION (PROOF OF MATCH)

The ultimate verification that the simulator aerodynamic math model is valid for a particular maneuver is a quantitative comparison between the simulator and the aircraft response. To perform such a match, the engineering simulator is driven by actual measured values of control surface position and engine thrust, as well as wind speed and direction throughout the time history. The simulator response is overlaid with the measured airplane time history.

For the above example, a simulator-to-flight comparison was made using a completely updated simulation except with predicted basic lift as shown in Figure 3. The resulting simulator-to-flight match is shown in Figure 5. The match is quite poor, and violates the tolerance of ± 3 knots airspeed¹. When the basic lift coefficient is corrected to that shown in Figure 4 using the value of ΔC_L extracted from flight data, the match is greatly improved as shown in Figure 6.

When the entire update process is completed, time history comparisons such as these, as well as snapshot data for steady-state tests such as longitudinal trims, are provided to the training simulator manufacturers. The resulting document may be used to aid the development of the Approval Test Guide which is presented to the regulatory authority as one requirement for simulator approval.

CONCLUSIONS

BCAG has long recognized the need for high-fidelity flight simulation both for flight-crew training and for engineering application. Standards for both applications are becoming increasingly stringent. The flight update of the aerodynamic mathematical model is key to producing a faithful representation of the airplane.

The method used at BCAG to extract aerodynamic coefficient data from flight-test measurements is successfully applied to all types of flight maneuvers. The method is effective for extracting data in the linear and nonlinear regions. The method is most effective when based on well-defined flight-test maneuvers which are performed in nearly smooth air. The updated mathematical model must be validated by a

comprehensive set of quantitative comparisons between simulator and flight-test results.

The quality of the flight-updated mathematical model is only as good as the accuracy of the flight data. Measurement of control surface positions and airplane kinematic data, such as rates, accelerations, and attitudes now is quite good. Improvements to the accuracy of engine thrust, mass data, flow angularity, pressure data, and winds could result in better simulator math modelling and validation results.

REFERENCES

1. Royal Aeronautical Society, "International Standards for the Qualification of Flight Simulators", draft, January, 1992.
2. FAA Advisory Circular AC120-40B, "Airplane Simulator Qualification", July 29, 1991.
3. International Air Transport Association, "Flight Simulator Design & Performance Data Requirements", 4th Edition, 1993.

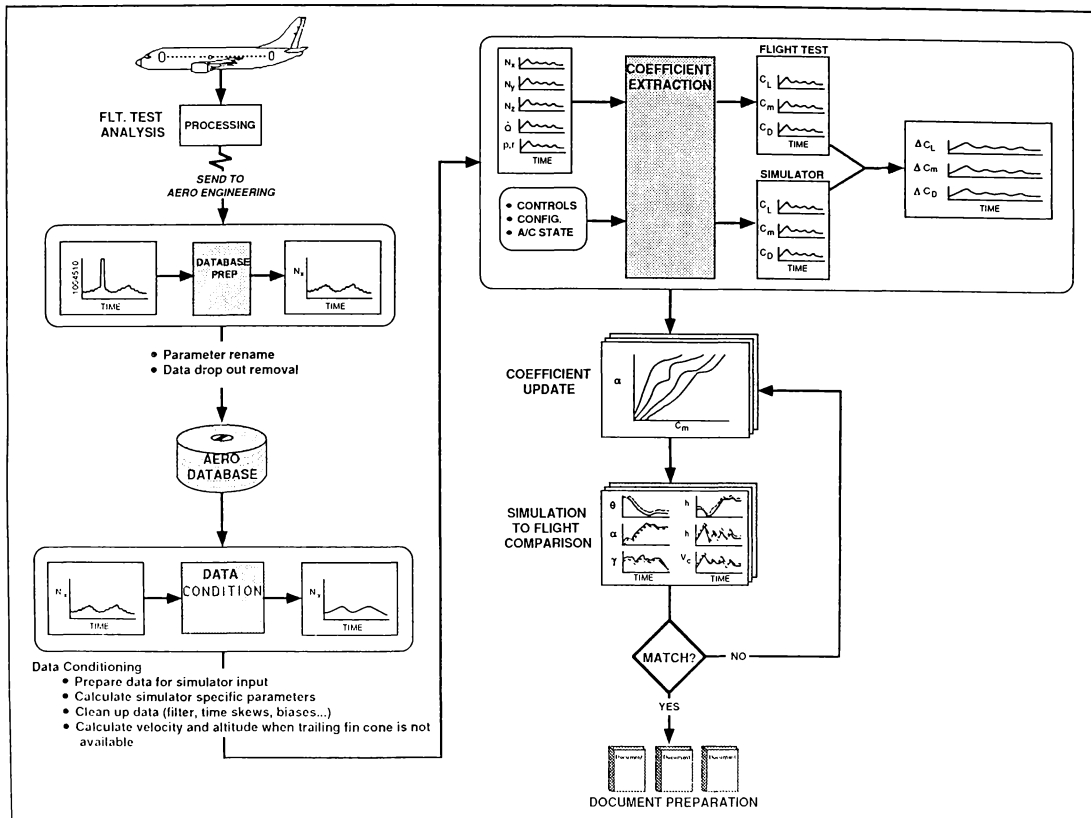


Figure 1. Aerodynamic Math Model Update Process

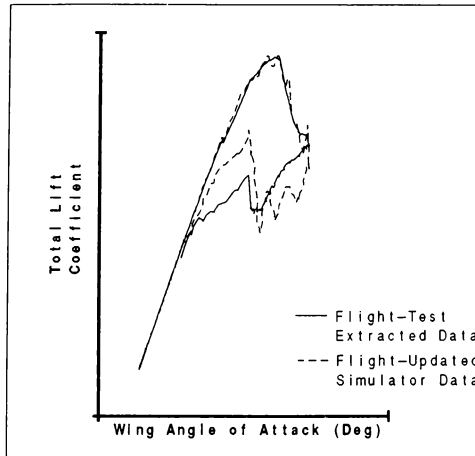


Figure 2. Flight-updated simulator total lift coefficient and extracted total lift coefficient for example stall test

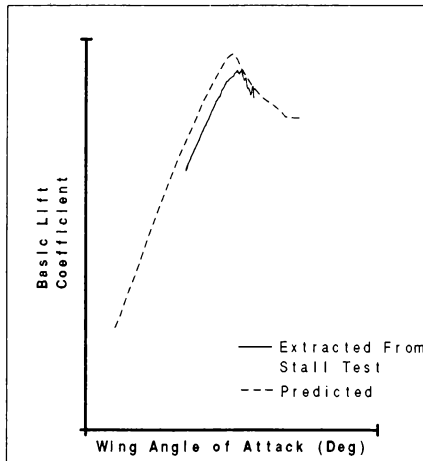


Figure 3. Predicted and extracted basic lift coefficient

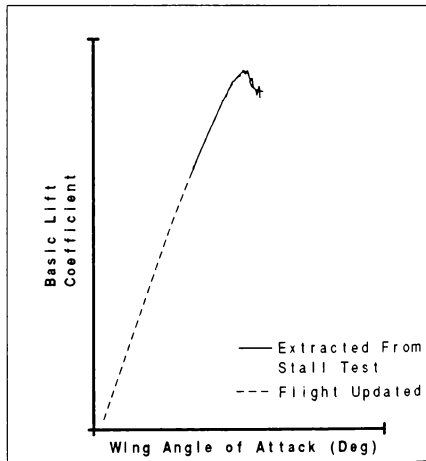


Figure 4. Flight-updated and extracted basic lift coefficient

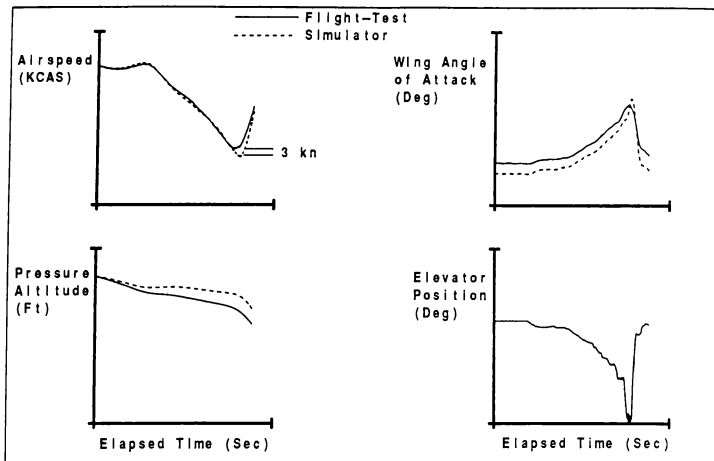


Figure 5. Simulator-to-flight match
Predicted basic lift

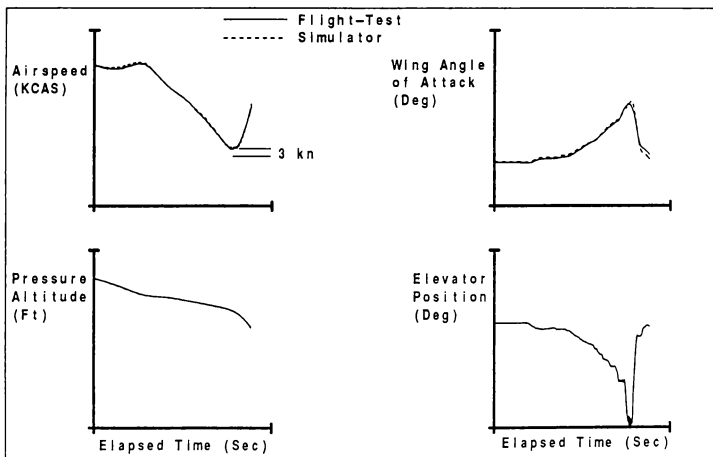


Figure 6. Simulator-to-flight match
Flight-updated basic lift

HOW TO CONSIDER SIMULATION FIDELITY AND VALIDITY FOR AN
ENGINEERING SIMULATOR

Zhang, Boding*
China Flight Test Establishment
Xian, China

Abstract

Simulation fidelity and validity for an engineering simulator have always had a strong appeal to the simulation specialists and customers. Along with enhancing the engineering simulator's standing in analysis and test task, simulation test and simulator design are emphasizing the importance of simulation fidelity and validity. Based on some methods in the process of integration, testing and acceptance for the SB-100 Flight Simulator and a comparison between simulation testing results and the results from flight, how to consider simulation fidelity and validity for an engineering simulator is discussed in this paper.

Introduction

Fidelity and validity in simulation for an engineering simulator are very important problems which require a great deal of attention by the simulation community. Until confirmed by flight testing the actual aircraft, manned flight simulation and simulation results in ground-based and in-flight simulators should be considered as predictions, at least. Although many studies have been performed over the years to compare simulation results to those of actual flight tests, to date there is still no absolute correlation.

This paper discusses simulation fidelity and validity associated with ground-based engineering simulation.

By simulation fidelity is meant the degree of closely representing the behaviour of a simulated aircraft by comparing the simulation environment and results in the simulator with those of the aircraft. The simulation fidelity includes the reality for the pilot feel and the correctness of the test results. Simulator fidelity can be broken down into the following parts: software fidelity (data and simulation models), hardware fidelity, fidelity for a whole tested system, fidelity of the pilot's subjective impression, simulation mission fidelity and simulation experience fidelity. Generally, the better simulator fidelity is, the better its validity is. Simulation validity is dependent on how much effectiveness of simulation tests, and the

extent or limitations for solving the problem of simulation engineering. It contains probability and coverage of solving simulation problems, the benefits after simulation tests, exercise and training, and effectiveness of providing with the information on the testing process.

Simulation validity (or effectivity) is its usefulness, namely creditability after tests in an engineering simulator, and it depends upon whether the test results could be available for use to good purpose and practical value of the flight simulation.

For a training flight simulator, there have been criteria for verifying the simulator in many countries (see Table 1), mainly, levels of approval proposed by FAA in the U. S and CAA in the UK. Many airplane companies have presented some methods to define the confidence for an engineering simulator. But there has not been a unified level of the engineering simulator approval validated by the proper authorities in the world.

Though the levels of simulator approval for a training flight simulator could give a reference to an engineering simulator, these two types of simulators are very different in their purposes.

For an engineering simulator, the grade of perfection and the technical requirements are not concerted at different development stages. According to the different test tasks in it, the simulation disposition has a little bit difference so that how to evaluate an engineering simulator is more complicated than a training flight simulator in a sense. In order to gain experience with fidelity and effectivity assessment, and to study methods of the increasing reliance upon manned engineering simulation for research and development, this paper first presents a technique view for simulation fidelity and then discusses test results performed in the SB-100 Simulator. The SB-100 Simulator is as a simulator with large angle of view and it can be flown near the flight envelope of a modern aircraft (from close ground to ceiling height with different altitudes, different air speeds and different engine conditions), and conducted aerial acrobatics and sustained maneuver. For further details, it can be found from

Basis of Simulation Tests

The high quality of flight simulation is greatly dependent on data provided to the simulation model, and technique and past experience of people who program and use the simulator. Flight simulation is not duplication, but 'an organic whole' which puts skill and art together. We have done a lot of work for building the simulator equipment and fulfilling the simulation tests in support of flight tests at CTFE over the past 30 years. Nowadays we should research into the simulation fidelity to improve simulation tests and enhance the effectiveness for use. In order to apply satisfactorily an engineering simulator to support the flight tests, six aspects of simulation fidelity should be mainly considered;

1. Simulation Fidelity

A. Simulation Hardware Fidelity

The performance of hardware in an engineering simulator can be compared to design specifications of actual aircraft systems and simulated flight environments so as to analyze and validate the test results. Sometimes the error required by simulator hardware is smaller than the manufacturing tolerance. Change of simulation hypothesis and expansion of operating limits are within the research scope of the hardware fidelity. The related problems of the hardware fidelity were discussed in reference 2.

B. Simulation Software (Simulation Models and Data) Fidelity

It deals with aspects of methods of mathematical modelling, computer programming, computational algorithm and check approach, of reliable source and lookup of simulation data, and of adequate representation and accuracy of the model and so on.

For simulation data, the requirement is necessary to enable an engineering simulator to adequately represent an actual aircraft. The exact requirements for data, in terms of scope and accuracy, depend on the particular research program and stage.

For simulation model, this model must possess a satisfactory range of accuracy, which can be achieved in real-time and which is acceptable and sufficient for the role of the simulation and can be in comparison with reality and consistent with its intended application. It also is in relationship with model validation, model verification and model specification.

C. Fidelity for a Whole Tested System

This system represents a simulator open-loop system. It is essential to compare the output results produced on a simulator with real-life results recorded from flight tests, to match the simulator response characteristics to the aircraft response characteristics, to make compensation for the simulated aircraft system according to the simulation requirements, and to achieve adequate accuracy which the real-time solution yields the steady-state performance of the aircraft and its transient behaviour.

D. Fidelity of the Pilot's Subjective Impression

That is the fidelity of man-in-the loop control. It means whether the simulator is like the aircraft, and deals with interface and relationship of man-machine, and pilot's subjective evaluation, experience, skill and knowledge.

E. Simulation Mission(or Task)Fidelity

It discusses the relations among test tasks, capabilities of simulator equipment and simulation requirements. The detail requirements of a given simulation task will vary depending on its purpose. For simulation task, we should understand that which device is first importance for the simulation test and which one is secondary importance.

F. Simulation Experience Fidelity

It indicates fidelity of judgment, experience, and intuition of simulator designers, simulation and test engineers and specialists. This fidelity seems to be out of simulator fidelity, but it's very important, otherwise if simulator users made mistakes in selecting the simulator equipment, a satisfactory simulation result can not be yielded in the simulator, even if it is a very good and expensive one.

2. Interrelationship among Test Tools

The interrelationship among test tools also belongs to the research scope of simulation task fidelity. In order to succinctly introduce the relationship among an engineering simulator and other test tools, their advantages, defects and relationships between outputs and inputs are shown in Fig.1.

Method of Study

To evaluate the simulation fidelity of the SB-100 Simulator, we usually use six methods for comparing the characteristics and performance of a simulator with those

of a simulated aircraft.

1. Man-in-the Loop Control Delays

Man-in-the loop control delays between pilot input and simulator response are major indexes of the flight simulator performance. It is one of parameters which has a serious influence on flying qualities for an aircraft. In past, because of the system of an aircraft is simple, even if there was a longer transfer time delay for the aircraft, the pilot could complete a low gain task with a near open loop. But for the modern aircraft of today, if there is a longer transfer time delay of its response, it is more difficult to complete a high gain task with a closed loop (as the pilot is a link in the large control loop). As a result, before simulation tests, the transfer time delay of a simulator must be determined in detail and then evaluation of flying qualities for an actual aircraft being simulated can be exactly performed in the simulator.

In general, the transfer time delay simulated in the simulator is longer than that of a simulated aircraft. Without changing aircraft basic performance (control force and aerodynamic characteristics in a simulator), it is necessary to reduce the transfer time delay. Measures used in the SB-100 Simulator are:

A. In order to increase reality conducted by the simulator, we utilized some unique technique including a multi-minicomputer network, a parallel computer processing method and a mixed language programming design used by different high-level languages so that the simulator is designed to have a maximum iteration time interval between 33.3 ms and 50ms depending on simulation tasks.

B. Make some compensation for the flight control system.

C. Improve the dynamic characteristics of the flight instruments.

D. Use the simple target tracking displays for some test tasks (without traditional visual systems).

Through determining the time delay for the simulator, the following experiences were gained:

A. The time delays contained in the FAA requirements are suitable for an engineering simulator. It can be one of the evaluation criteria for an engineering simulator. An important factor is the difference between the simulator delay and the simulated aircraft delay.

B. During performing the software processing, specially using the compensating measures, the basic performance, aerodynamic characteristics

and flight control systems for an aircraft can not be changed. Generally speaking, the damping of a flight control system is smaller and the bandwidth of it is wider. The time delay of the whole control system against total simulator delay is smaller (about from a thirtieth to a fourtieth of the total delay). Therefore, more effective method is to raise the iteration computation in terms of high computer speed.

C. Different test conditions will get different test results. It is necessary to select reasonable test conditions and decide the detail test requirements in different flight stages.

D. During process of measuring the transfer delay, before tests, the simulation test has to keep the simulator systems constant (i. e. initial test conditions in steady balance states).

E. The repetitive error of simulation results for each test is very large (because of many links and systems in the simulator). The measuring method and the same testing repeated times for defining the simulation delay should be also considered. For example, the smallest throughput delay from the input of the control stick (pitch channel) to the output of visual displays in the SB-100 Simulator is 117 ms, but the average throughput delay is 163 ms. It is shown in Fig. 2.

F. In order to exactly determine the time delay, the best way is to use a high-speed camera with time coordinates.

G. To check influence of delays on flying qualities, it can be determined by comparing ground-based simulation with in-flight simulation.

2. Total Output Results

Total output results based on the aerodynamic parameters were measured in the SB-100 Simulator. Besides measuring the frequency and the damping ratio in each channel (including the damping ratio of longitudinal short-period, Dutch-roll frequency and damping ratio, roll control performance, roll control response sensibility and control sideslip amplitude and lateral mode characteristics). System transient output response to the control stick (or pedal) inputs recorded of the simulated aircraft must be compared with those from flight tests. Based on the comparative results to correct aerodynamic data and simulation models so that the matched error is as small as possible. The relative matched error should be within the range of ten (or larger) percent for an engineering simulator. In this study,

the test regimes and input modes are conducted in the same way as those used in the study in actual flight test. The inputs (on control stick or pedal) may be taken from a tenth to a fifth of the total displacement, using a single pulse wave for the pitching system, a double pulse wave for roll and yaw systems. The comparison between simulation results and flight test results is available from Ref. 3.

3. Analogue Input Errors

The analogue inputs mainly are the inputs of all the control systems and the inputs of the engine thrust levers. Accuracy of the analogue inputs will directly influence on simulation fidelity and simulation test validity (specially high requirements for the test tasks of precision control and close formation). Thereby, to enhance simulation fidelity, errors of analogue inputs (including simulation resolution, quantization error, 'fluctuation' error of analogue inputs as a function of time) must be controlled critically. The high-amplitude 'fluctuation' error (zero-drift error) is thrown into confusion so that the pilot is difficult to control the instantaneous value of analogue inputs and even could not keep the instantaneous balance state for the simulated aircraft. The time histories of stabilator input errors and throttle input errors were measured on the SB-100 Flight Simulator, and are shown separately in Fig. 3 and Fig. 4. In the former case, the error is larger than the allowable tolerance which is intended. In the other, the error is smaller than the allowable tolerance.

The following measures must be employed to reduce this error:

A. Utilize a high precision potentiometer (separate-level type or multiturn).

B. The distributed capacity and their interference should be dealt with carefully, specially, correctly handle ground strap among all the simulator equipment.

C. Use accurate A/D channels (16 bits or higher) and reference supply with high stability.

D. Correct the 'fluctuation' errors by software methods.

4. Various Errors in Displays

During flight simulation tests, one of the pilot's main information provided by the simulator is a visual cue. It includes a visual (SKY-Earth) reference in three degrees of unrestricted rotational freedom

over the entire field of view, a target movement and displays, the horizon and targets belong to simulation of external environments. Under existing conditions, there has not been an efficient and entirely quantitative method comparing with its objective reality so that these performance determination and correction are only dependent on the evaluation which the pilots present during conducting different flight maneuvers, and by reference to associated standards. But displays belong to part of simulated aircraft systems. It can make a performance comparison between simulated displays and actual aircraft displays. In system demonstration tests, the indication error is measured by the following ways:

A. Utilize three different input modes---two static measuring methods (based on equal angle input and real standard scale input) and one dynamic measuring method.

B. Take mean error, RMS error and relative error of displays in both direct and reverse movement (see Table 2), and neutral position (zero) error, total mean error, repetitive error, sensitivity and response speed (see Table 3) and absolute error in specific points of displays (such as Table 4), by use of the static measuring method.

C. Determine dynamic characteristics of displays by use of the dynamic method.

D. Measure throughput delays from the input of the control stick to the output of main instruments (including HSI, g-meter and altitude rate meter). We measured performance of all the instruments in order to compare the performance of simulated instruments with those of true instruments and examine the static and dynamic performance of the simulated instruments according to the requirements for flight tests, and then found which performance parameters can be had influence on assessment of aircraft flying qualities and why they may be had influence on flying qualities. After flight simulation tests, the pilots will offer evaluation and suggestions for an improvement on the flight instruments. In terms of making a comparison between objective, quantitative analysis and subjective assessment, we can easily find out reasons why these instruments are in accord with the requirements and some others are out of accord with the requirements. In general, the indication error of simulated displays should be smaller than those of actual aircraft displays. Because only in this way, they could be made compensation for a difference between the simulated system

and real aircraft system, and be used for engineering simulation.

5. Typical Maneuvers

After subsystem testing, system testing, systems integration, evaluation testing and acceptance testing, specially through determining transfer delays, time transient curves, analogue input errors and indication errors of displays, main problems existed in the simulator and the major difference between the simulator and the actual aircraft can be found out so that how to improve the simulator may be further defined. In order to enhance the fidelity of the simulator, the methods of hardware improvement and software compensation are used. But how to prove that the engineering simulator is entirely useful for support of the flight test, we have to do a lot of work for the simulation validity. The following flight tests have been completed on the SB-100 Flight Simulator;

A. Basic Aircraft Performance

- . Maximum level-flight velocity
- . Level-accelerating flight and level-decelerating flight
- . Climb and descent performance
- . Maximum vertical velocity (maximum rate of climb) and ceiling
- . Aircraft operating envelope

B. Aircraft Control and Stability

C. Typical Maneuvers

- . Steady turn with different banks
- . Zoom flight and diving
- . Tooth method flight
- . Roll (full and half)
- . Coordinated turn
- . Wind-up turn
- . Climbing turn
- . Half roll and split-s
- . Loop
- . Target tracking

Based on the performance determination and test data, models of aerodynamics and data of engines can be calibrated, and simulation test results can be compatible with those from flight by repeating the iteration number of times (see Table 5). six simulation tests are listed in Table 5 in terms of difference between simulator testing data and data from flight. If we just conduct a few flight validation tests, such as takeoff, landing and cruise to judge the accuracy of software models and data, it is not good enough for a highly-maneuverable aircraft. We must know which range of data and models are right and which range of them are wrong so that the previous simulated flight tests are entirely necessary to evaluate and validate the engineering simulator. The

value of this work lies not only in helping to validate simulation models and simulator data, it also helps to set simulator standards for some research tasks.

6. Pilot's Evaluation and Suggestions

Pilot's evaluation and suggestions provide the primary basis for improving the fidelity and validity of a simulator. According to their experience of practise and deep comprehension on flight, the pilots indicate sharply the symptom of the simulator and where is noncoincident with the flight rule, and which parts of the simulator are different from those of the simulated aircraft (where is the difference in, how much difference, what is real circumstance, what reason of the difference can be estimated, and how to improve etc.).

6.1 Evaluation

In order to have pilot's evaluation on simulation fidelity, the items evaluated by the pilots are split into three sections as follows;

. Evaluation of typical flight subjects and comparison with actual flying tests

. Employing opinions on major systems (Earth-Sky scene, target, control feeling and intercoordination among them) offered by the pilots

. General comments given by the pilots

6.2 Suggestions

After repeating tests, the pilot's suggestions can be concluded as follows;

- . Reality of Earth-Sky scene
- . Target tracking
- . Control (or environment) feeling
- . Dynamic response
- . Precision control
- . Observed reference during aerob-

batic maneuvers

Due to limited space, it is unnecessary to go into details.

Conclusion

In order to enhance the effectiveness for use, it is necessary to reduce errors of the simulator systems. Hence we pay more attention to the precision of hardware and software for the simulator, and made a comparison between simulation testing results and the results from flight. After numbers of tests and analyses, major sources of simulator fidelity problems can be found out, we try to study verification and validation methods, and criteria, and to gain quantified system analysis for manned engineering simulation fidelity by

measuring attributes of piloting technique in the simulator, analyzing the errors for the simulator hardware and software, and comparing the simulation testing results with those measured in flight tests.

Conclusions reached are:

This paper mentions the relationship between engineering simulator testing and other simulation testing approaches and their influence on fidelity of simulation.

There must be some different methods in fidelity study for different types of aircraft and for different testing tasks.

It represents points of view and approaches for efficiently enhancing simulator fidelity and effectivity.

It introduces some ways of improving simulation fidelity and validity for the SB-100 Flight Simulator.

References

- [1] Edward R. Jones, The Interrelationships Between Engineering Development Simulation and Flight Simulation, 50 Years of Flight Simulation Conference Proceedings, Section 3, 1979.
- [2] Zhang, Boding, SB-100 Simulator Design, Features and Associated Problems Study, Proceeding of 2nd Beijing International Conference on System Simulation and Scientific Computing Volume 1 1992.
- [3] Yu Zhi-gang, Research Works on Fidelity of Flight Simulation in CFTE, CFTE 1992.
- [4] J.M.Rolfe, K.J. Staples, Flight Simulation, Cambridge Aerospace Series, 1986.
- [5] Simulator Census, International Flight, 1991.3.

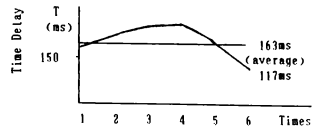


Fig.2 Transfer Delay from the Input of Control Stick (Pitch Channel) to Output of Visual Displays in the SB-100 Simulator

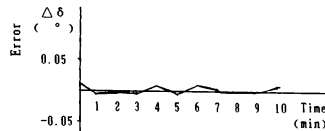


Fig.3 Errors of Stabilator Analogue Input

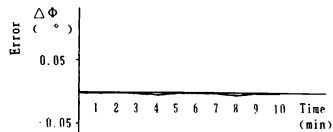


Fig.4 Errors of Throttle Analogue Input

Table 2 Indication Errors of Displays

Linkage Channel NO.	Instrument Name	Measuring Method	Parameter	Error		
				Mean Error	RMS Error	Relative Error(%)
1	Revolution meter (right)	Equal Angle Input	n Direct	0.73	0.86	0.69
			n Reverse	1.27	1.01	1.18
2			n Direct	1.00	0.91	0.80
			n Reverse	1.50	1.02	1.43
5	Accelerometer	Standard Scale Input	+ny Direct	0.01	0.03	0.10
			+ny Reverse	0.03	0.04	0.30
			-ny Direct	0.02	0.03	0.20
			-ny Reverse	0.04	0.04	0.40

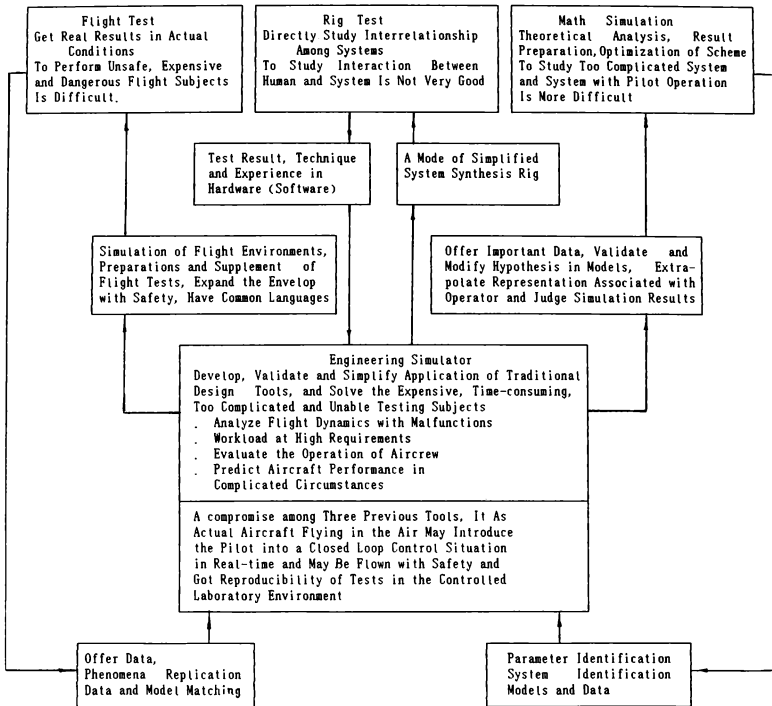


Fig. 1 The Relationship Among an Engineering simulator and Other Test tools

Table 1 Differing National Simulator Criteria

Country	Authorized organization	Certification level
USA	FAA	Level A
		Phase 1-level B
		Phase 2-level C
		Phase 3 level D
UK	CAA	Level 1-5
Australia	ACAA	Level 1-5
Germany	LBA	Phase 1-3
Japan	JCAB	Phase 1-3
Norway	NBA	Phase 1-3
China		Level 1-4

Table. 3 Dynamic and Static Parameters of Displays

Instrument Name	Parameter	Total Mean Error		Repetitive Error		Sensitivity		Response speed	
		Desired Value	Measured Value	Desired Value	Measured Value	Desired Value	Measured Value	Desired Value	Measured Value
Airspeed Meter	Indicated Velocity(km/h)	15.0	12.5	10	5	5	1	100	490
M Meter	M Climbing(m/s)	0.02	0.02	0.01	0.01	0.01	0.01	0.1	1
Climb Meter	and Diving Speed	2.0	0.9	1.2	1.2	1	<0.1	25	>100
Yawmeter	Magnetic Heading (°)	±1	1	±0.7	0.7	±0.5	±0.5	30	30
HSI	Pitch Angle(°)	±0.5 (as 0°)		±0.4					
		±0.7 (as ±30°)	<0.5	±0.6	<0.3	0.3	0.3	60	60

Table. 4 Comparison with the Desired Value and the Measured Value for HSI(Pitch Angles.°)

Test Point	0	5	10	20	30	40	60	80	-5	-10	-20	-30	-40	-60
Allowable error	±0.5	±0.7	±0.7	±0.7	±0.7	±1.0	±1.0	±1.0	±0.7	±0.7	±0.7	±1.0	±1.5	±1.5
Measured error	0	0	0	0	0	0	-0.1	-0.5	+0.5	+0.5	+0.5	+0.5	0	-0.5

Table. 5 Comparison With Simulator Test Data and Data from Flight

Test Item	Condition	Difference between Simulator Testing Data and Data from Flight			
	Altitude(km)	Min. Indicated Velocity		Max.	M
Aircraft	3	+9.6		-0.004	
Operating	5	+2.7		-0.005	
Envelope	10	-0.9		-0.015	
	15	+6.0		-0.004	
	Altitude(km)	M		Climb Velocity(m/s)	
Rate	5	+0.034		+1.4	
	(Full Power)				
of	9.3	-0.030		-0.6	
Climb	5	-0.019		-0.7	
	11	-0.035		-4.4	
Climb Time	Altitude(km)	time (s)			
	1~20	7			
Ceiling		Altitude (m)			
		400			
Turning	Altitude(km)	M	Normal(g)	time(s)	
			g Loading		
Performance	5	-0.033	-0.2	+0.6	
Loop		Altitude (m)	g Loading (g)	Indicated Velocity (km/h)	Path Angle(°)
	Entry Condition	-200	-0.8	-45	0
	Top Parameter	+335	-0.58	-42	0
	Exit Parameter	+200	+0.3	+22	-15
	time		no difference		

FUTURE MILITARY PILOT TRAINING

- A PERSPECTIVE OF INDUSTRY -

Dieter Illauer
LtCol. ret. GAF, Consultant
Deutsche Aerospace AG
Munich, Germany

Abstract

Military pilot training is faced with severe constraints and restrictions in the future. Therefore effective training with the use of simulation by application of new technologies is gaining increased importance.

Extensive studies and research have been performed by Deutsche Aerospace (DASA) for several years for the German Ministry of Defence and the European Fighter Aircraft Consortium on subjects such as pilot training analysis, Tactical Simulation Center, contracted training and joint multinational training. As a result, with regard to the changing training scenario, an Advanced Training Concept for future military pilot training has been developed by Deutsche Aerospace. It aims to accomplish the training tasks by increased and optimized use of synthetic training aids in order to substitute parts of flight training and by participation of industry in the field of training. By these means, and by restructuring the training program, life-cycle-cost can be reduced and the effectiveness of training improved at the same time.

This paper outlines the Advanced Training Concept and gives a perspective for future military pilot training. It describes the training problem, with the training need and the constraints for training. Also covered is the role of simulation for training and available technologies. Conclusions for future training are derived. The presentation is the view of industry, specifically that of Deutsche Aerospace, Germany.

Introduction

My presentation outlines the view of industry, specifically that of Deutsche Aerospace, for an Advanced Training Concept for future military pilot training. Deutsche Aerospace conducted extensive studies in the recent years for the German Ministry of Defence (MOD) and for the European Fighter Aircraft Consortium on subjects, such as pilot training analysis, Tactical Air Simulation Center, contracted training, and joint multinational training.

First, I will try to identify the problems of future military pilot training by pointing out the training need and the constraints. Then I will address the role of simulation as a tool for training and advanced training technology.

Deriving from that, I will present the Advanced Training Concept and present a perspective, on how Deutsche Aerospace thinks the future training problems of the Air Forces can be solved.

The Training Problem

The Training Need

Some 20 years ago prior to introducing the F-4F, the German MOD performed manned A/A combat simulations comparing the F-104 vs the F-4F, and vs the different threat aircraft. During the tests, we soon found out, that we had to rotate the pilots and everybody had to fly every aircraft under all conditions. This was because simply one good pilot or one mistake in air combat could spoil the whole result. The assessment later showed that pilot proficiency overrules aircraft performance: A well trained pilot in an inferior A/C will win against a badly trained pilot in a superior A/C !

The job of training military combat forces is one of the most important and demanding of all military requirements. Especially in the last years, dramatic changes in the world situation occurred with engraving impact on military operation and training.

These changes are: 1. Political reorientation of some Nations, especially the former USSR, Warsaw Pact Countries, and Germany, in conjunction with strategic changes in the world wide threat. 2. Regression in the economics with cut backs of defense budgets. 3. Fundamental changes in the role of the forces of some Nations, e.g. Out-of-Area missions, overseas missions and changes in force structure. 4. Rapid changes in technologies.

In the subject of military pilot training, we first have to clarify what training in this context means:

The Air Forces have a need 1. to train aircrews entering the Forces after being qualified as pilot candidates (Re-generation Training) 2. to convert experienced aircrews from one aircraft type to another, normally to limited Combat Ready status (Conversion Training), and 3. for continuation training of the pilots in the operational wings to achieve and maintain a Combat Ready status in their specific roles (Tactical Continuation Training).

The training objective for the Military is, in short, to get the aircrews Combat Ready and to keep them proficient at the highest possible standard in the shortest time with the minimum of cost.

This task is becoming more and more difficult. Worldwide on the potential threat side, especially in the Far and Near East, more capable SAM Systems and Fighter Aircraft are being introduced. On the other hand in the western nations, more sophisticated aircraft are being brought into inventory which are probably easier to fly than old F-104s or F-4s. However, they require more training to fully exploit the edge of the modern weapon systems against the adversaries.

Aircraft with extensive avionics and weapons have become more complex and sophisticated than ever before. The scenario has changed and areas of operation have increased, for some nations beyond the borders of home defence. Pilots must be more flexible by flying in several roles and operating in different geographical areas at varying environmental conditions.

Constraints for Future Training

Training of military aircrews is subject to numerous restrictions and limiting factors which result from the political and economic situation of a country and from the mission of the forces and flight safety aspect.

Six main factors and trends can be identified:

1. Reduced number of personnel.

The diminishing threat to the European Nations from the East and Disarmament Agreements of the past are leading to a reduction of armament and forces, which primarily will result in less military personnel in the future.

Also, some Air Forces are faced with the problem that qualified personnel for highly specialized and demanding assignments have to be recruited from an ever decreasing number of candidates.

2. The limitation of funds.

In the future, an ever decreasing defense budget will have to be expected. If the steadily increasing operating costs are added to the cost increase for the procurement of weapon systems, the financial margin will be further limited. As a consequence, the need to reduce operating costs must be given greater consideration when developing future training plans and selecting of training aids.

3. Lack of realistic training conditions.

In most of the countries many of training objectives cannot be achieved by live training with the aircraft, because realistic training conditions, such as realistic ground and air threat, "real" ECM-environment and employment of modern weapons do not exist.

Also, for reasons of security or cost, certain tactical missions and events cannot be trained live.

Rapid development of technology will accelerate and extend air combat and air battles and will lead to increasingly complex scenarios which will exclude live training within the limited airspace available.

4. Enhanced flying safety requirements.

For peacetime operations, flight safety must rank first: Any mission that might likely result in hazards to the health or lives of people cannot be trained with full realism. For some Air Forces it will be necessary to look for training alternatives particularly for low-level flight and air combat.

5. Heightened environmental awareness.

An increasing public interest in environmental protection and in the reduction of aircraft noise requires some Air Forces to significantly limit its training operations. The flying units are compelled to cut down on their training programs and/or to avoid certain airspace. Therefore, Air Forces of the European Nations have to accomplish some elements of their training programs overseas at high cost.

6. Longer in-service life of the weapon systems.

In view of the high development and procurement costs, all Weapon Systems must be designed for a long in-service life and be operated sparingly reserving them for combat. For this reason, training missions with the operational aircraft should be limited to what is absolutely necessary.

In spite of the need and the growing importance of training, a negative trend can be observed in the training tasks actually being accomplished: 1. the flight hours and missions per pilot are steadily decreasing, for some high value A/C already below the proficiency minimum. 2. the number of exercises flying are reduced from around 10 per year in the sixties to 1 or 2 per year currently. 3. live firing and weapon delivery training have been severely cut down as a result of few remaining firing ranges and drastically increased cost of guided ordnance.

In summary all Air Forces have drastic training deficits today and they will be further aggravated in the future.

The Role of Simulation

In view of these constraints, the question arises, how can simulation contribute to the accomplishment of the military pilot training in areas such as: 1. training of pilot candidates 2. conversion training of aircrews 3. continuation training of aircrews 4. evaluation of tactics and procedures 5. practice of war roles and 6. mission rehearsal.

Simulation Today

Today various types of flight simulators are being used in the Air Forces for different functions of pilot training. Generally, the simulator training serves as a supplement and ranks second to flying training, as shown with the following numbers which may be considered representative for most of the Air Forces:

Flight Hours	Simulator Hours	Ratio
per pilot / per year		
approx 180	approx 18	10 : 1

It can be stated, that the potential of simulation for aircrew training has not been utilized to capacity in the past:

Essentially, this can be attributed to the following factors: 1. there were no technologies available that would have allowed realistic simulation 2. the degree of acceptance of the systems was low 3. simulation was of secondary importance, in particular as far as tactical continuation training was concerned

4. fuel cost and flight hour cost were low in the past 5. enough funds were available to cover the cost for flight hours and practice weapon deliveries 6. there were little restrictions for flying operations and 7. the environmental awareness of the population and the media was not yet sharpened.

Simulation in the future

In the future simulation must focus on the following objectives:

1. Improvement of the proficiency of the pilots.
Exercises and experience clearly show that by effective utilization of synthetic training aids, the performance of the aircrew can be improved substantially.
2. Reduction of training times
The utilization of simulators for training purposes will result in a reduction of overall training times.
3. Reduction of personnel
Particularly of supporting elements. The amount of possible reduction will depend on the degree of simulation utilization and/or how many flying will be substituted by simulator training.
4. Reduction of training costs
Comparison between cost per flight hour and cost per simulator hour for modern aircraft reveal ratios from approximately 10 : 1 to 5 : 1. With more sophisticated simulators (high resolution visual systems, complex scenario capability etc.) the ratio could further decrease, but would still be in favour of the simulator.
5. Reduction of hazard of live flying.
The training of critical situations and conditions in missions are possible. Hazards to the population and to civil aviation can thus be avoided.
6. Environmental relief
The training of maneuvers which present an environmental strain (low-level flight, supersonic flight, air combat) would be minimized with the use of simulator.

Besides these overall objectives, which may vary in priority for some European Nations, there are specified objectives and training trends for the use of simulation in the future.

Generally, future weapon systems require a higher quality and a greater amount of simulation.

The training of all kinds of fighterbomber type aircraft with low-level missions are hampered by the noise problem. The GAF will take extensive measures to fly a high portion of these missions in the simulator in the future.

The tactical training program of air defense aircraft will have to predominantly be trained in the simulator because of noise nuisance by supersonic phases.

For cost reasons, the future training of air-to-ground and air-to-air weapon employment with expensive dispenser and guided missile type weapons, will be mainly restricted to simulators.

The training of electronic/combat reconnaissance aircraft with their mission spectrum, including electronic warfare within the scope of Suppression of Enemy Air Defense (SEAD) and weapon employment for anti-radiation type weapons must be accomplished exclusively in the simulator.

While training of these tasks in the simulator was not realistic or in some cases not possible in the past, future flight simulators will allow effective training of complete tactical missions in complex scenarios.

They not only improve the proficiency of the aircrew but also reduce training cost considerably. Additionally, as illustrated in Figure 1 a "Force Multiplication Effect" can be achieved by properly complementing aircraft training with effective simulator training.

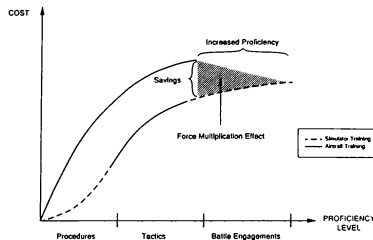


Fig. 1 Simulator Effectiveness

Advanced Training Technology

To achieve the training objectives advanced techniques and technologies must be relied upon.

For the future, the following trends are apparent:

The utilization of electronic training aids can obsolete the use of mechanical training aids and the traditional classroom instruction. It will be possible to tailor instruction to the individual requirements of the student and reduce the training period, thereby improving the total training capability.

New technologies for flight simulators provide a variety of options for realistic simulation of out the window visuals, motion, sound and scenario generation as well as instructor stations.

The generation of terrain, radar, and infrared computer generated image (CGI) data with photographic quality resolution has become possible and will become cheaper in the future. A high resolution of the visual system is the key to realistic simulation and a prerequisite for any substitution of live training.

Reconfigurable cockpits, where the instrumentation the performance and flight characteristics can be modified by means of software (depending on the type of aircraft to be simulated) will make it possible to simulate several types of aircraft in a single simulator. This will reduce expensive hardware change of cockpits, particularly during the development phase of new weapon systems.

Modular configurations allow standardized interfaces between training systems and within the simulators. These hardware and software modules which can be made portable, can be used for various weapon systems, various training equipment and support equipment.

Because of the low hardware expenditure required, interactive control stations are considered to be a cost-effective solution for providing manned stations and to conduct interactive multiple air combat operations.

Embedded training (i.e. a simulation capability integrated in the original equipment) is a possible and attractive way for future simulation. As an integral component of all weapon systems, it could contribute to a solution where expensive live targets or even certain simulators might be eliminated altogether.

The present solution and the handling of the various decentralized data bases by various countries and industry, as well as various simulator technologies, are not standardized and thus not as cost-effective as possible. It is therefore imperative to standardize the databases at an international level, as a matter of urgency.

Creating a database library, containing world wide and correlated terrain-, radar- and IR-data base available for all nations on a lease basis, should be a long-term objective.

Battle level simulation, by means of using multiple interactive tactical simulators, or linking simulators over a short distance within a Tactical Air Simulation Center or over large distances at operational wings and command posts, allow training, exercises and mission rehearsal in a synthetic or a manned battle environment.

In this context, I like to stress that 1. new technology is not a substitute for training analysis; it just expands the options for implementing training, and 2. training requirements drive the use of technology; it should not be used just because it exists.

Direction for the Future

Changes in defense scenarios, increasing financial and environmental constraints, and advanced technology will change the requirements of the world's Air Forces. Military pilot training will be dictated by the need of 1. improved training effectiveness 2. lower training cost, and 3. total training (see Figure 2).

On the basis of these factors, Deutsche Aerospace is presenting the following main conclusions by offering an Advanced Training Concept. It can be considered to be the perspective and direction for military pilot training in the future.

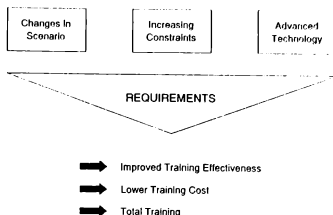


Fig. 2 Factors Influencing Future Training

The Advanced Training Concept consists of the following areas by which the training objectives of various nations could be achieved: 1. increased utilization of synthetic training aids; 2. optimum mix of training aids; 3. re-structure of training programs; 4. joint procurement of training aids; 5. joint training; and 6. participation of industry in training.

These subjects will now be covered in more detail.

First: Increased Utilization of Synthetic Training Aids

Historically, simulator training served as a supplement to actual flying and no significant cost saving effect was attained because of the high investment cost for simulators. In the future, flight hours should be substituted by simulator hours to produce overall savings.

As pointed out before, insufficient training capabilities, safety regulations, the lack of training areas and environmental aspects simply make it necessary to shift an increasing portion of training tasks from actual flying to simulation, which is especially true for those portions of training which cannot be practiced live anymore.

By training analysis regarding to which parts and to what extent the training programs can be substituted and by utilizing simulation properly, a higher training effectiveness can be achieved with lower cost. This is schematically illustrated in Figure 3.

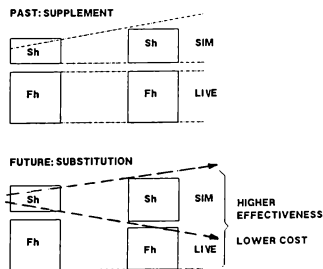


Fig. 3 Simulation vs Live Training

Of course, there will never be a total substitution, however, a lower annual flight hour requirement of the training programs could be the result.

As an example, Deutsche Aerospace analyzed the tactical training program of the German Air Force (GAF) with several options of varying numbers of flight hours to be replaced by simulator hours at a ratio of 1 : 1. This step-plan is illustrated in Figure 4

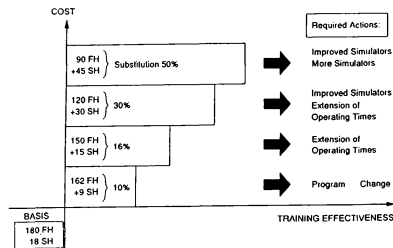


Fig. 4 Substitution of Flight Hours by Simulation

Also the relevant impacts on cost saving and effort required was studied with the following results: 1. a replacement of 10 percent flight hours by in-service simulators will be possible short-term by only changing the training program; 2. 16 percent of flight hours can be shifted to simulators by increasing the daily operating times of the simulators; 3. to allow 30 percent of the flight hours to be replaced by simulators, both qualitative improvements of the in-service simulators and around-the clock simulator utilization will be necessary; 4. a conceivable shift of 50 percent of the flight hours to simulators would require comprehensive qualitative improvements of the in-service simulators, the introduction of new simulators and the establishment of a Tactical Simulation Center.

When considering the shifting of flight hours or training tasks to simulators, 3 types of options have been identified:

First: there are training activities which only can be accomplished in the simulator (e.g. because of restrictions due to airspace, tactics, reconnaissance/intelligence security, flying safety aspects, cost and environmental aspects). The analysis of the tactical training programs of various Air Forces revealed, that approximately 30 percent of training tasks of future tactical aircraft with a recognizable increasing trend can be trained only in simulators.

Second: there are some training tasks which can be flown both in the simulator and in the aircraft. This amounts to approximately 40 percent of the training program.

Third: there are training events that must be flown live as a matter of course (i.e. all flying activities where human factors such as courage, fear, stress, risk, decision making, familiarization,

weather, g-loads, etc. are involved). This part consists of around 30 percent of the training.

For the step-plan, as shown before, a cost model was applied to calculate the annual savings by substituting flight hours by simulator hours with the following result: 1. if 10 percent of the flight hours of the tactical training program are substituted by simulation, the savings will amount to approximately 100 Million DM per year. 2. if 50 percent of the flight hours are substituted there would be an annual saving of approximately 400 Million DM. These figures refer to the GAF with TORNADOS and F-4s and 600 aircrews flying 180 hours per year, and include the investment cost for the required increased simulator capability.

The prerequisite, however, of moving into the direction of substituting flight hours is a high level decision by the Airstaffs. This would also be dependent upon a program to engure acceptance by the aircrews.

Second: Optimum Mix of Training Aids

For the future we see the need of introducing an optimized spectrum of training aids to complement each other (see Figure 5):

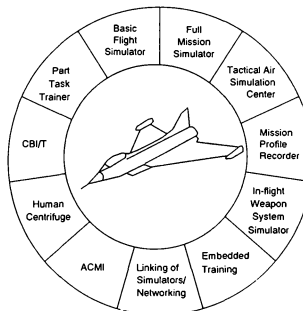


Fig. 5 Training Aids Spectrum

Computer Based Introduction/Training (CBI/T) as an interactive display workstation for all kinds of academic training

Part Task Trainer (PTT) particularly a cockpit trainer and an avionic trainer, to train new or converting students in all cockpit functions and procedures, specifically in handling the avionics.

Basic Flight Simulator (BFS) as the primary simulator for training new and converting pilots in handling of the aircraft, instrument flying, navigation, formation and emergency procedures.

Full Mission Simulator (FMS) with a capability to simulate complete tactical missions including interactive air combat, low-level flight and weapon delivery with 2 or more aircraft in a mission specific scenario.

Tactical Air Simulation Center (TASC)
There is an operational need for a central simulation facility for the training of high value missions involving more than 6 participants in a complex tactical scenario. It is to simulate the operational employment of all types of weapon systems in air warfare and is the only system to permit the training of joint air operations. The TASC is a long-term project. It could be a joint venture for the nations participating in the European Fighter Aircraft (EF2000) Program or for other European nations. To save cost, it should be established using already existing simulation facilities available at industry or within the Air Forces.

Linking of Simulators and Operational Units

In the future linking of all simulation systems among themselves or with operational units, including command and control posts, is technically feasible. Such linking would allow the conduct of large-scale exercises, however, would also need a central management and control station.

Embedded Training

which provides the pilot with the capability to fly against simulated targets in a simulated threat environment using his own aircraft.

In-Flight Weapon System Simulator

Onboard simulation of sensors, avionics and other systems is considered to be an effective and cost-saving way to relieve the use of costly weapon system equipment for advanced training.

Mission Profile Recorder

A mission profile recorder, like the US system ACTESS (US) or the COMTESS (GE), comparable to ACMI, however, as a range independent device, is to record the conduct and the success of a mission. It is the required for mission rehearsal and qualitative training.

Air Combat Maneuvering Installation (ACMI)

is a valuable means to train and to assess pilots. We see a need to establish one or two in Central Europe besides the existing ones at the North Sea, and at Sardinia, Italy.

Human Centrifuge

is highly desirable to train aircrews periodically for their operation under high for g-loads in a realistic cockpit environment.

Primary Training System 2000 (PTS2000)

Rockwell International and Deutsche Aerospace are teamed to compete for the Joint Primary Aircraft Training System (JPATS) with a recently developed aircraft, called "RANGER 2000", which presently is being tested in United States and Germany. Several nations plan to replace their present basic trainers and use JPATS (including aircraft, simulators and ground school) to train pilots in basic flying.

Advanced Training System 2000 (ATS 2000)

Deutsche Aerospace is working on a project study of an advanced trainer aircraft, called ADVANCED TRAINER

2000 as a contender for the replacement of present advanced trainers.

AT 2000 will have supersonic performance with an extended flight envelope and modern avionic systems to cope with the advanced flight training requirements of the year 2000 and beyond.

Alternative Aircraft

As a result of our training analysis, a portion of the annual flying training program of the operational wings could be flown with another aircraft (non-operational type) to save valuable and costly flight hours in the operational role-type aircraft.

The Advanced Trainer 2000 would ideally be capable to fulfill this secondary role.

Third: Re-Structure of Training Programs

In regard to substituting live flying with simulation and the utilization of the broad spectrum of training aids, the training programs need to be modified.

To train the candidate pilots from the start to a limited combat-ready status we recommend a Training Pipeline. It consists of an optimized build-up training in complementing aircraft parallel to academic and simulator training using complementing synthetic training aids (see Figure 6).

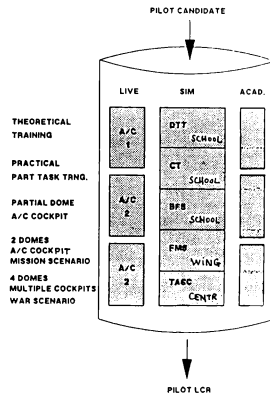


Fig. 6 Training Pipeline

For continuation training of the pilots in the operational wings to achieve and to maintain a combat-ready status, an optimized mix of all training facilities should be used, including the aircraft.

The simulation portion consists of 1. Computer Based Instruction/Training; 2. Full Mission Simulator and; 3. Tactical Air Simulation Center.

The live training comprises the flying of specific missions with the alternative aircraft and flying tactical missions with the operational aircraft 1. in the unit; 2. at tactical ranges and; 3. in combat exercises (e.g. RED FLAG) (See Figure 7).

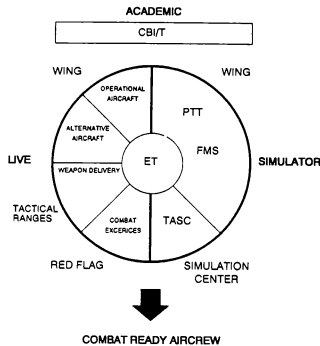


Fig. 7 Continuation Training

While in the past the tactical training program of aircrews in most of the Air Forces had to be considered as quantitative (where only number of flight hours counted), in the future pilot training must be orientated qualitatively with regard to less and more valuable flight hours. No time should be spent in any training setting (classroom, simulator or aircraft) without having defined and measurable objectives. For qualifications and licence renewals simulator training should be used as equivalent to live training.

To ensure these objectives, tactical training programs should be restructured as illustrated in Figure 8:

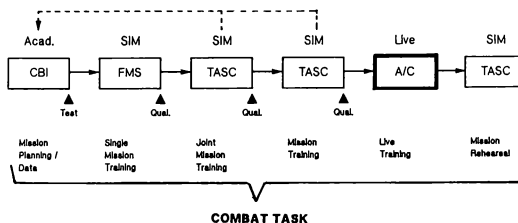


Fig. 8 Future Tactical Training

Training should be split in complete combat tasks rather than in events, such as e.g. "Attack of an Airfield with an Aircraft Package". After theoretical mission analysis and planning, the mission should first be flown single in the Full Mission Simulator, then single in the Tactical Air Simulation Center and then with the aircraft package under the full scenario in the Tactical Air Simulation Center. Only after having demonstrated the required success the mission should be flown live. Between the different phases, qualifications would have to be passed by the aircrew.

We also see a need to accomplish mission rehearsal and to have the ability to perform war roles which are not possible in the aircraft in peacetime. Missions and total air operations such as those during the Gulf War or over Yugoslavia should be practiced in synthetic interactive battle environments before sending the aircrews on their actual mission.

Fourth: Joint Procurement of Training Aids

In order to aim for the highest possible level of cooperation and standardization, we see the need for joint procurement of training aids.

Joint ventures for design, development, production, integration and support of military equipment is not only a NATO Council policy, but also have been practiced among nations and industry with mutual benefit. Joint procurement of aircrew synthetic training aids for the European Fighter Aircraft is also a strategy of the European nations involved.

This procedure has the advantage of 1. enhancing the political, economical and military cooperation; 2. having identical standards, and; 3. achieving the most cost-effective solution for the nations.

To fulfil the objectives of reducing cost and assuring cost-effective solutions, we suggest the following procurement philosophy for training aids: 1. simultaneous development of aircraft and training aids; 2. maximum commonality between the aircraft and the simulators; 3. applying a modular design for HW and SW around a common CORE, and; 4. using the "best available proven technology" to reduce risks.

Fifth: Joint Training

Joint training has been used by some nations for years with great success. I would like to mention 1. the joint F-104 training at Luke AFB, US; 2. the Euro NATO Joint Jet Pilot Training (ENJJPT) at Sheppard AFB, US; 3. the GE/IT, UK joint TORNADO training at the TITE Cottesmore, UK; 4. the multinational tactical training at Goosebay, CA, and; 5. the multinational weapon employment training at Decimomannu, IT.

Also, EF-2000 joint training is envisaged by the four nations Germany, Italy, Spain and United Kingdom. The participating companies Alenia, British Aerospace Construcciones Aeronauticas, Deutsche Aerospace) recently produced a Pre-Feasibility Study in order to provide consolidated information about possible training options.

Two options have been studied in detail: 1. national training, whereby all aircrew training is done at national training units in each country, and 2. quadrinational training, where the aircrew training is done at one site. This is the so-called Four Nations EF-2000 Training Establishment (FETE).

It was shown, that a joint quadrinational solution has the following benefits: 1. enhancement of European cooperation; 2. high degree of standardization and training effectiveness; 3. reduced need for military personnel because of contracted elements or, possibly the whole training; 4. infrastructure and facilities to be provided only by the host nation; 5. least training restrictions by weather, airspace, noise, etc., and 6. minimum training cost for the nations because of cost sharing.

As a perspective, this joint training center could be the basis for all primary and advanced flight training in Europe as a long term solution.

Compared to this solution, the option of independent national training revealed many distracting disadvantages: 1. it is a step away from political, economic and military cooperation; 2. standardization and training effectiveness is limited; 3. each nation

requires full military personnel; 4. each nation will have to provide and support an airfield and also respective training areas; 5. training will be restricted by the national constraints, and 6. training cost for each nation will be higher.

To highlight this cost aspect, a comparison between FETE vs 4 national training units showed a saving of approximately 50% for each nation, or vice versa, there would a cost increase of approximately 100% for each nation if they would decide for independent national training. In these figures the build-up of the training center, 8 years of operation, and procurement cost for simulators are included.

Also, it has been assumed that approx. 80% of the services at the training center would be contracted to industry. Such a percentage is considered feasible.

The training would be under the aegis of military command. The military would mainly be responsible for flight operation, administration of a small military staff and the provision of instructor pilots. All other services, like: 1. academic and simulator training; 2. the provision of flight ready aircraft, including maintenance, logistics and; 3 all general base facility services and supporting services, would be the responsibility of the contractor.

Besides these two main options other options of operating a joint training center have been studied. They cover cover the spectrum from "total military operation" to "all industry operation" (See Figure 9).

Special emphasis was given to the study of: Option 3: industry owns simulators, aircraft ground equipment (AGE) and provides services; Option 4: industry owns simulators, AGE, A/C spare parts and provides services, and Option 5: in which also the aircraft belong to industry. Training aids would be either leased or a total training package in terms of a trained/ converted pilot would be sold to the Air Forces.

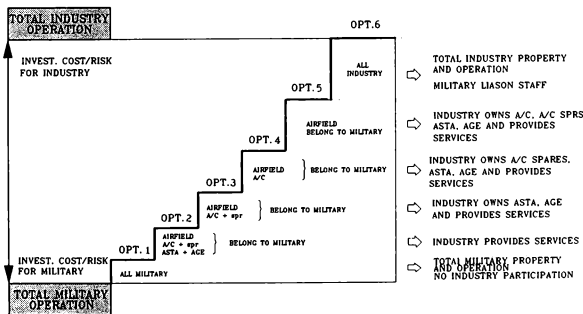


Fig. 9 Options of Operating a Training Center

On one side these options help the Air Forces to solve peak funding problems by reducing the initial procurement cost. On the other side, they will lead the possibly unacceptable high flight hour costs, which are increasing for Option 3 : from approx. 25.000 DM; Option 4 : approx. 50.000 DM; Option 5 : to approx. 100.000 DM per FH. These figures include 40 A/C and 100.000 flight hours over 8 years operation.

Sixth: Participation of Industry in Training

Contracting of Training

Military aircrew training was, historically seen, first a commercial task which shifted to government services almost exclusively especially after World War II. Since about 1980, a shift back to contractor managed aircrew training can be observed. There are highlighting examples on this subject, such as: 1. Contractor-managed Training Programs and; 2. Total Contractor Training (TCT), besides ATS which is solely used with contractor-managed academic and simulator training.

Especially in the United States, the trend is apparent towards more contractor and less military involvement in almost every aspect of aircrew training, simply driven by manpower and budget constraints.

Total Contractor Training is an approach to acquire aircrew training, whereby the level of training and the desired learning outcome is specified rather than the hardware. The contractor provides or specifies what equipment is necessary to achieve the desired learning outcome. The government would specify number of crews to be trained, the skill level of the entrants and the required skill level or qualifications of the student at training completion.

Contracted training is beneficial to both, the military and the industry.

For the military: 1. experienced manpower can be transferred to operational assignments; 2. lower cost potential for system acquisition and fixed operating cost; 3. training would be more efficient and effective as it is done by highly qualified personnel being steady in the job; 4. the load of capital investment is spread over a long period, and 5. the technical, schedule and financial risks are imposed to industry.

For the Industry: 1. training extends through the service-life of the aircraft (vs. a 2-3 year training aid program in the past); 2. there are economic advantages for the nations and industry; and 3. there is also a benefit by the dual use of development hardware and software.

Establishment of a European Training Company

In light of the training need and the constraints with which the Air Forces are faced in the future, we are convinced that the aerospace industry should team

in order to provide not just aircraft, but integrated total training - instead of training equipment as in the past.

The product "Total Training" would comprise: 1. provision of aircraft, aircrew and ground crew training devices, software, course ware, data management systems; 2. the build-up and operation of a multinational training center and; 3. logistic support, including services for maintenance and repair of aircraft and training aids (See Figure 10).

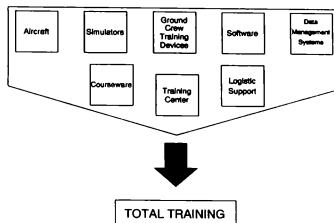


Fig. 10 Elements of Total Training

The overall management responsibility for the total weapon system, including training, should reside at one company. It should be stressed that the provision of training is a task for the aircraft companies and not for the simulator companies, as only the aircraft manufacturers are capable of providing total training in a cost-effective manner.

In our opinion, only centralization assures: 1. maximum commonality of equipment; 2. maximum efficiency for design, development and manufacture of all training aids; 3. design-to-minimum life cycle cost; and 4. an overall maximum cost effective solution for the Air Forces.

We therefore see the need for the foundation of a new European training company, let me call it "EUROTRAINING" for the time being.

The beginning could be a training company consisting of the four EF-2000 partner companies ALN, BAE, CASA and DASA for EF-2000 training. In this case by providing total training EUROTRAINING would complement the work being done by the companies Eurofighter and Eurojet. As a result the EF-2000 nations would be in the favourable position to receive the aircraft and, at the same time the required number of pilots trained to limited combat ready status on the EF-2000.

Concluding Remark

In order to meet the training requirements under the constraints imposed, the military will have to investigate new ways of fulfilling their tasks in the future. They will have to modify and augment today's training methods with highly sophisticated training aids in a computer generated synthetic battle level training environment.

Our perspective for the future is a training environment with the following elements:

1. Severly reduced flight hours and live-exercises for training with a high amount of synthetic training to fill the gap;
2. Synthetic interactive battle-engagement environments used as primary means of training operational tasks;
3. Live training only to enhance the proficiency;
4. Increased joint ventures for procurements of training systems and joint multi-national training and;
5. Increased industry contribution in training, in some areas, providing "Total Training".

We think that this Advanced Training Concept of the Deutsche Aerospace offers to the Air Forces a solution to their training challenge.

USAF IN-FLIGHT SIMULATION:
A COST-EFFECTIVE OPERATING APPROACH

Steven R. Markman
In-Flight Simulation Program Manager
WL/FIGD
Wright-Patterson AFB OH
45433-7505

ABSTRACT

A major factor in the success of the U.S. Air Force's in-flight simulation program has been keeping operating costs low while retaining needed skills, capabilities, and operational flexibility. A suitable management approach and operating concept are critical to the success of this program. The Air Force has developed, owned, and used in-flight simulators since the early 1950s. The current aircraft, the NT-33A and the NC-131H, are operated as national resources and are available for use throughout the U.S. Government, U.S. industry, and foreign allied governments. The nature of the program often requires quick response and the ability to adapt to the needs of the user. The current operating concept is Air Force management and ownership of the aircraft, contractor operation by a highly specialized contractor, and a tasking type contract on which to employ the contractor. This operating approach has proven to satisfy all of these requirements. This approach has evolved over the history of the program to make the aircraft available to users and to insure valuable and cost effective results.

INTRODUCTION

The Flight Dynamics Directorate of the Wright Laboratory, which is a part of the Air Force Material Command, has developed and operated in-flight simulators since the early 1950s. The current USAF in-flight simulators are the NT-33A (Fig 1) and the NC-131H (Fig 2). The VISTA (Variable Stability In-Flight Simulator Test Aircraft) NF-16D (Fig 3) is currently being developed to replace the NT-33A and is expected to begin research and development flight operations by the end of 1994.

The effective use of in-flight simulators requires a management approach that allows flexibility and quick response while keeping costs as low as possible. Without this management approach, the full technical capabilities of the aircraft cannot be realized.

The specific means of managing and operating in-flight simulators has evolved over nearly forty years. They are now operated as a national resource and are available for use by the Air Force, Navy, Army, NASA, U.S. industry,

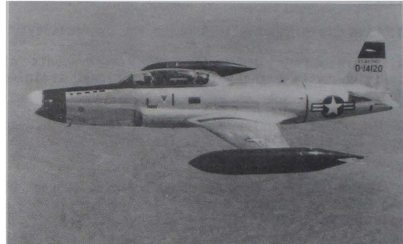


Fig 1 - NT-33A



Fig 2 - NC-131H Total In-Flight Simulator (TIFS)

and foreign allied nations. Operating in-flight simulators requires very specialized skills. The necessary mix of engineering, piloting, technician, and maintenance skills needed to operate one-of-a-kind in-flight simulators has always dictated a contractor operation. The need to respond quickly to a diverse range of customers requires a flexible contracting approach that is difficult to accomplish in today's contracting world.

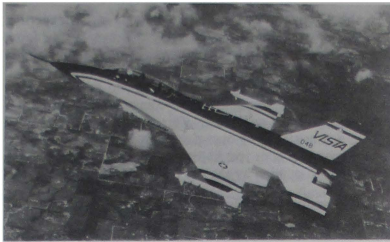


Fig 3 - Variable Stability In-Flight Simulator Test Aircraft (VISTA)

FLYING REQUIREMENTS FOR IN-FLIGHT SIMULATOR AIRCRAFT

The types of unique research flying that in-flight simulators perform must be understood in order to comprehend why such a different management approach is needed. The USAF uses in-flight simulators to support four general types of efforts:

1. **Pre-First Flight Evaluation of New/Modified Aircraft** - This use gives the program most of its reputation. However, this type of simulation is actually a small portion of the total use because the U.S. military does not develop new aircraft every year. Simulations have included bombers such as the B-1 and B-2, fighters such as the F-15, F-16, F-18, JAS-39 Gripen, Lavi, YF-22, and YF-23, research aircraft such as the X-15, AFTI/F-16, and X-29, and the Space Shuttle.

2. **Research and Development** - This is the mission for which in-flight simulators were first developed and operated. They are ideally suited for flying qualities research and many other R&D efforts because the flight characteristics can be varied easily and the effects of new controls, controllers, or control mechanizations can be demonstrated to test pilots and flight test engineers. This is especially true for evaluating new flight control concepts, developing flying/handling qualities specifications, and testing displays and human factors concerns.

3. **Specialized Training** - USAF in-flight simulators provide training at the Air Force and the Navy test pilot schools. This flying accounts for about half of the flying hours, keeping the total flying hours up resulting in decreased costs for all users. The NT-33 is used to demonstrate a wide variety of flying qualities and HUD formats. In a one and one half hour flight, about thirty different combinations of aircraft dynamics, stick dynamics, control system feedbacks, time delays, and HUD formats are demonstrated. The NC-131 is used to train pilots and engineers to test avionics. Test techniques are taught in a comfortable but real airborne environment. In this role, the NC-131 is configured with radar, FLIR, and E/O sensors, and INS, GPS, and LORAN navigation systems. Many internal parameters of these avionics systems can be accessed through a special computer and data bus.

4. **Flying Laboratory** - In-flight simulators make excellent flying laboratories. The aircraft are already highly instrumented, and the procedures for designing and approving the installation of new or special test equipment are well established.

CUSTOMER REQUIREMENTS

The Air Force has chosen to operate its in-flight simulator aircraft as national resources, making them available to other agencies within the U.S. Government. This includes the Navy, Army, NASA, and the FAA. They can also be made available to U.S. industry and foreign allied governments.

The need for the contractor to be independent of any airframe manufacturers or control system designers has long been established. While a manufacturer who already has a large flight operations department could probably develop the expertise to operate in-flight simulators, programs often involve proprietary or competition sensitive data. Thus it is felt that only an operator who is independent of any manufacturing concerns can provide a credible evaluation, often of competing products.

Most R&D programs are closely tied to the availability of funds. Sometimes the sponsor is able to insure funds availability well in advance, but many times they are provided on short notice. This is especially true when a problem is discovered in a new aircraft and it is decided to use in-flight simulation to investigate. In a situation like this when quick response is essential, it is impractical to write a new contract for each test effort to be performed. Thus it is essential to maintain an ongoing contract on which funds can be obligated quickly to start the contractor working in a timely manner.

It can be said for any research or test facility that its reputation is only as good as its last effort. In the Air Force's case, that reputation has developed over nearly forty years. Failure to detect a major flying quality deficiency could do damage to the program's credibility that would require years to repair. Thus, consistent, high quality results are essential for every program performed. This consistency is best achieved by insuring a continuity of personnel involved in every aspect of managing and operating the program.

OPERATING CONCEPT REQUIREMENTS

The effective use of in-flight simulators requires that they be able to respond quickly to help resolve technical problems. This demands that an ongoing capability exist, one in which qualified people who know how to program, fly, and maintain the aircraft be available. It requires being able to respond to the priorities and needs of the program sponsor, being able to fly when they want and at the location they want. It also requires a contractual vehicle that allows the contractor to begin working quickly and has general enough wording to allow for unplanned and unanticipated uses of the aircraft.

OPERATING CONCEPT

The concept for operating the in-flight simulator aircraft has evolved over many years of operation. It consists of four elements:

1. Government ownership of the in-flight simulator aircraft
2. Contractor operation of the aircraft
3. Basing the aircraft at the contractor's facility
4. Tasking type contract

GOVERNMENT OWNERSHIP - A question often asked is "Why not give or sell the aircraft to the contractor and then hire them as needed to perform specific simulations? This would free the Air Force of management responsibility for the program". The above statement is true, but there are other advantages to Air Force ownership.

Although there is a mechanism to either sell or give the in-flight simulator aircraft to the contractor, this would require the contractor to insure the aircraft and obtain liability coverage. This would certainly be very expensive given that the aircraft are highly modified and perform research flying. This extra cost would be added to the cost to operate the aircraft. The best solution is for the Air Force to retain ownership of the aircraft and provide them to the contractor as Government Furnished Property. The contractor is freed from most liability associated with their approved operation of the aircraft by accepting Air Force oversight on their maintenance and operations.

The Air Force has rights to all data collected and all reports prepared for every program. This data can then be distributed throughout the Government as needed. The only exception is if the data is classified or procurement sensitive and special arrangements are made in advance. Thus the program office becomes a clearing house for data generated using the in-flight simulators.

Last, by owning the aircraft, the Air Force has control of a precious national resource, allowing it to establish goals and set priorities. This gives the Wright Laboratory many "intangible" benefits. The aircraft are a visible sign that the laboratory is aircraft oriented, not limiting itself to laboratory tests and computer analyses. New technologies can be tested and demonstrated in an airborne environment early in their design, giving the laboratory visibility especially within the flight test community. The aircraft also are a means by which laboratory personnel can maintain contact with other flight test, design, and engineering agencies within the U.S. Government, industry, and foreign governments.

CONTRACTOR OPERATION - The question is often asked "Within the entire U.S. Government, isn't there anyone who can operate the NT-33, NC-131, and NF-16 in-flight simulators?" The answer, of course, is yes, but the Air Force chooses to have a contractor operate the aircraft for a number of reasons.

First, and probably most important, is the expertise that has been established by the contractor over nearly forty years of designing and operating in-flight simulators. Several of the engineers who participated in the original development of the NT-33 in the 1950s and the NC-131 in the 1960s are still working for the company are now senior level managers and engineers. This continuity of personnel insures a consistency of operation, aids in faster training of new personnel, adds credibility to research results, and insures a high opinion by the rest of the test community.

Generally, a contractor has greater flexibility and responsiveness to hire and utilize personnel with unique combinations of skills than does the government. They also can hire retired military personnel and put them into engineering and flying positions, whereas the military tends to put such senior people primarily into management positions. Government agencies tend to categorize personnel into specific skill categories and often lacks the flexibility to take full advantage of all their capabilities. A small contractor usually has that capability. For example, all contractor pilots who fly the in-flight simulators are also engineers. In addition to flying duties (which can include flying as either a safety or evaluation pilot and being qualified as a safety pilot in more than one aircraft), pilots also perform program set-up, data analysis, and program management duties.

Having the flexibility to mix personnel as needed for any given task allows "cross fertilization" between specialties. Nobody works just NT-33 or just NC-131. This goes for pilots, engineers, technicians, and maintenance personnel. Also, no one works just R&D programs or just test pilot school training. This allows for lessons learned from R&D efforts to be incorporated quickly into training flights for the test pilot schools, improved designs for the simulation system on one aircraft to be evaluated for use on the others, and the workload to be better spread out when the aircraft deploy to a remote operating location.

Also, this cross fertilization keeps personnel from becoming stale or burned out from doing the same thing day after day. This can be seen in the enthusiasm the contractor always displays and is a major factor in their ability to retain personnel for twenty or thirty years or more.

CONTRACTOR OPERATING BASE - The contractor provides the support base where programs are set up and the aircraft are maintained and prepared for test programs. Being capable of an autonomous operation enhances the overall flexibility and responsiveness of the program. The sponsor has the option to fly their program at the contractor facility or at some other location dictated by their program requirements. Being capable of an autonomous operation and not dependent on the priorities of a host base allows the program to deploy to a remote operating site that may have only basic services available.

Another advantage of an autonomous contractor operation at their own facility is that programs can be flown with little or no visibility to the remainder of the flight test community. This enhances the capability to perform classified simulations. Operating in this manner where the in-flight simulator is normally seen allows such programs to be performed in relative anonymity.

TASKING TYPE CONTRACT - The Air Force has very strict rules for the wording contained in contracts and for procedures for contracts to be awarded. Although there are different types of contracts to give various levels of flexibility, every contract has to specify four items: (1) the work to be performed, (2) the product to be delivered, (3) the cost, and (4) the schedule. Also, the funds, or a schedule by which funds will be obligated on the contract must be provided. A major conflict of requirements exists here; little, if any, specific R&D work is known at the time the contract is written. There are no funds provided, and no schedule and end product specified.

The contract has to comply with laws and regulations that do not always match well with the requirements for operating in-flight simulator aircraft. A close and open working relationship between program management and contracting personnel is essential to establish and maintain a contract that meets procurement requirements while still providing the flexibility needed to operate the aircraft.

The solution to the above dilemma is called a tasking contract. Its key features are that only general requirements are spelled out in the statement of work, and then amendments or tasks are added one at a time as specific work to be performed is identified and funds are provided. Thus, the in-flight simulation contract states that the contractor will provide a facility, maintain the aircraft in a flight-ready status, and provide pilots and engineers capable of operating the aircraft. It also spells out in general terms the types of R&D and training efforts to be performed, but these examples are carefully worded to avoid limiting the contractor to only those types of efforts.

When a potential sponsor wishes to perform a program, they meet with the Air Force program manager and the contractor to determine the requirements for that program. The contractor then prepares a Task Plan which is essentially a statement of work for this effort. It contains the work to be performed, the cost, schedule, manpower, and Air Force support necessary to perform the effort. When all three parties are satisfied with the document, the customer provides the funds to the program manager. The Task Plan and funds are then given to the contracting officer who obligates the funds and issues a modification to the contract. The contractor can now begin work on the task and charge to the contract. This entire effort to prepare the task and start the contractor working can often be performed in one month or less.

Another big advantage of a tasking contract is that if there is no work to be performed (which occasionally happens), the Air Force is not obligated to pay the contractor. Knowing that funding is not guaranteed, the contractor has an incentive to keep the aircraft busy and becomes a partner in sharing the success or failure of the program. Although they cannot charge marketing activity to the contract, the contractor actively markets the capabilities of in-flight simulators to government and industry with visits, phone calls, technical papers, and presentations.

The Air Force is justly an advocate of full and open competition to find a qualified contractor. In most cases there are many sources to solicit and the Air Force encourages competition to lower costs and encourage creativity. In the vast majority of cases competition will help achieve these goals. However, there are many cases where there is only one source qualified to perform the work, or only one that can perform the work in a satisfactory manner in the required time frame at an acceptable cost (both financial and technical). The Air Force recognizes this and allows for sole source procurements, but it requires approval from a very high level with a large amount of justification (the current contract, which was awarded sole source, took nearly two years to prepare).

SUMMARY

The present in-flight simulation operating concept of Air Force ownership and contractor operation of the aircraft with the Air Force providing overall program management has proven to be a very successful concept. It allows for the needed flexibility at acceptable costs while assuring high quality research results.

Over the years through which the in-flight simulation program has evolved, a team approach has developed between program management, the contracting office, and the contractor. Each is aware of the requirements of the other and works toward the common goal of providing a flexible in-flight simulation capability.

A RAPID PROTOTYPING SYSTEM FOR INFLIGHT SIMULATION USING THE CALSPAN LEARJET 25

S. A. Bueche
Engineering Test Pilot
P. R. Deppe
Engineering Test Pilot
Calspan Corporation
Buffalo, New York

Abstract

Inflight simulations of new aircraft and/or flight control systems often require considerable preparation time and expense. Our objective was to develop a new inflight simulator with a "quick response" capability. The Calspan Learjet 25 inflight simulator has been upgraded to meet this objective through implementation of a graphical block diagram programming system with automatic code generation. The simulation code is targeted to a set of floating point digital signal processors (DSP) which operate in parallel. The system is designed to be capable of full flight-envelope partial model-following and employs new features to reduce checkout and calibration time. These include computer-flown test input sequences and on-line parameter identification. This paper describes the aircraft, its software development system, real-time computer hardware, user interface, and advanced inflight simulation and calibration methods. A "quick response" inflight simulation capability is now available to the aerospace industry which may be used as a tool in the development of new aircraft and flight control systems.

Introduction

The Calspan Variable-Stability Learjet 25 (Figure 1) is a production aircraft which has been extensively modified for use as an inflight simulator.



Figure 1. Calspan Learjet Inflight Simulator

The right seat controls (elevator, aileron, and rudder pedals) have been removed and replaced with variable electrohydraulic feel systems. The simulation system incorporates a digitally-controlled analog

computer similar to that used in Calspan's other Variable-Stability Learjet.

The aircraft was upgraded in the fall of 1992 with an additional digital computer system. The upgrade was designed to provide improved capabilities to rapidly set up and calibrate new aircraft simulations and to allow complex simulations not possible with the analog system. The upgraded simulation system is shown in Figure 2.

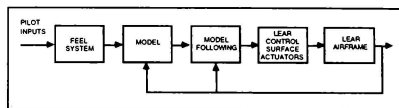


Figure 2. Inflight Simulation System Block Diagram

A simulation customer provides a model of the aircraft and flight control system to be simulated in the form of equations and block diagrams. This model is programmed in graphical block diagram form on a desktop computer. The desktop model is capable of non-real-time simulation and can produce computer generated "C" language code targeted to the real-time computers onboard the inflight simulator.

The evaluation pilot flies the simulation from the right seat by making inputs through the programmable artificial feel system. These inputs are fed into the model and model-following system which command the Learjet's hydraulically actuated control surfaces to move in a way which causes the Learjet to produce the motion of the simulated aircraft.

Computer Hardware

The digital computer system (Figure 3) is hosted in a 20-slot passive Industry Standard Architecture (ISA) backplane. The host processor is a single-board 33 MHz 80486 computer with 8 megabytes of memory and standard serial and parallel ports.

Mass storage is provided by an internal 90 megabyte Bernoulli removable-cartridge disk drive. A single cartridge is used for storing all inflight software, setup files, and flight data.

Three DSP subsystems operating in parallel are used for the bulk of the real-time computation. Each is based on the Texas Instruments TMS320C30 33 MHz floating-point DSP which is capable of 33 million floating-point operations per second.

The first DSP subsystem is manufactured by dSPACE GmbH of Paderborn, Germany. The three-

board set consists of a 32-channel, 16-bit analog input board, a DSP board, and a 5-channel, 16-bit analog output board. The boards are connected by a 32-bit Peripheral High-Speed (PHS) bus which allows the subsystem to operate autonomously from the host. The dSPACE system is typically used to model and control the variable-feel system, as described in a later section.

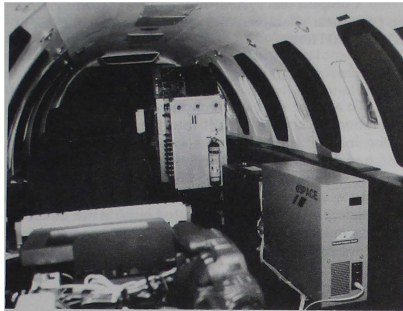


Figure 3. Simulation Computer

The two remaining DSP subsystems are manufactured by Dattel, Inc., of Mansfield, MA. Each consists of a multifunction card with 16-channel, 12-bit analog input and a TMS 320C30 DSP. A single Dattel 16-channel, 12-bit analog output card is shared by the two DSP subsystems. The Dattel systems are typically used to implement the model and model-following flight control laws, also described later.

Programmable display capability is provided by a high-speed graphics card driving an active-matrix color 640 x 480 flat-panel LCD display. The graphics accelerator card, by ATI Technologies Inc. of Scarborough, Ontario, Canada, is designed to perform graphics line drawing and clipping in hardware. The computational power of the DSP boards in conjunction with the fast graphics hardware enables the system to generate complex head-down displays at refresh rates of up to 100 Hz.

Software Development

The software development environment is based on Simulink, the graphical block-diagram-oriented interface to Matlab from The Math Works of Natick, MA. This widely-used commercial package has recently been enhanced to produce C language code appropriate for real-time execution directly from a block diagram.

Using Simulink on a Sun, Macintosh, or PC computer, the user constructs a hierarchical block diagram of the system to be simulated. The Simulink environment is similar to an object-oriented drawing program. One advantage of this approach is that once the system is entered graphically, it not only provides quality documentation, it also becomes a working model which can be analyzed on the desktop using the extensive

tools available with Matlab such as time histories, frequency responses, and pole-zero plots. This allows a model to be quickly debugged and verified for proper implementation and consistency.

Once the model has been validated, the Simulink code generation feature is used to generate real-time C code for the digital signal processors onboard the inflight simulator. The source code is cross-compiled on a desktop PC and linked with hardware-specific interface code. The resulting executable code is saved on a 90 MB Bernoulli cartridge disk, ready for use on the Learjet.

User Interface

The flight test engineer station (Figure 4) is equipped with a flat screen LCD terminal, keyboard, and trackball.

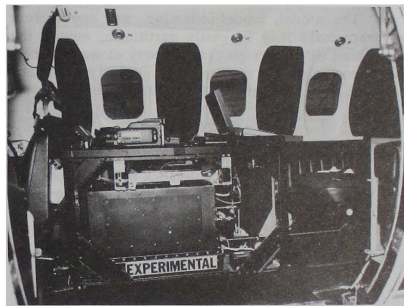


Figure 4. Flight Test Engineer Station

The main simulation control program runs under MS-DOS on the host processor and consists of a window-and-mouse-based user interface. The operator has access to all of the variables in the simulation software by name through dual-ported memory on the DSP boards. This capability allows monitoring or changing any model or feel system characteristic at any time during a simulation. The same Matlab script files used to set parameters in the desktop simulation are used to set parameters in the real-time code in the aircraft.

The control program also provides a variety of engineering displays. Eight time history traces (with overlay capability) are available as well as X-Y plots. Any variable in the simulation code may be plotted in real time. Display parameters may be changed interactively or saved and recalled from plot setup files.

The user interface program allows the operator to set up and control 128 channels of direct digital data recording. Any variable in the simulation code may be recorded in real time for postflight analysis.

Model-Following System

The aircraft model-following is an implicit three axis (pitch, roll, yaw) system. This means that measured Learjet states are used to compute model accelerations. The Learjet is then forced to have the same accelerations as the model. The model is considered to be implicit because there are no explicit integrations performed in the model to produce explicit model states. The system automatically compensates for Learjet fuel burn by computing the moments of inertia and center of gravity and using these quantities in the model-following equations.

Thrust sensors installed on each engine allow the Calspan Learjet 25 to model thrust effects of engines mounted in any location on the simulated aircraft. A recent flight test program used this capability to evaluate techniques for emergency flight control using thrust only.

The model, model-following, and any custom software tasks may be easily partitioned among the available DSP subsystems in a variety of ways to best meet specific project requirements. Typical simulations require a cycle time between 5 and 10 milliseconds.

Feel System

The artificial feel system is hydraulically powered and electrically varied. A centerstick or wheel/column and sidestick is available. The centerstick is shown in Figure 5.

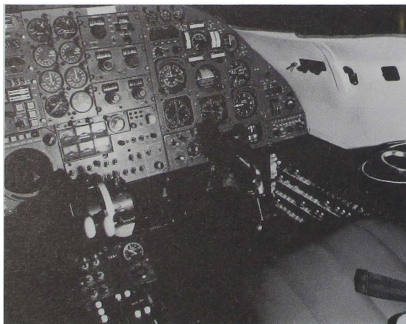


Figure 5. Programmable Feel System

The feel system is simulated using an explicit model-following technique as shown in Figure 6.

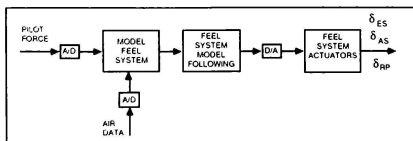


Figure 6. Feel System Block Diagram

The explicit feel system model is programmed using the block diagram code generation techniques discussed earlier. Characteristics include multiple nonlinear springs and dampers, variable mass properties, static and dynamic friction, preloads, freplay, aerodynamic reversibilities, failure modes, and artificial speed scheduling devices such as a "q bellows".

Complex feel system simulations, including all of the non-linear characteristics mentioned above, cycle at 2500 to 5000 Hertz.

Ground Simulation Capability

To aid the simulation checkout process, a ground simulation capability is built into the system. A single button in the Learjet activates a set of relay controlled switches to select simulated sensor inputs from a standalone ground simulation computer in place of the actual sensor inputs. The ground simulation computer is a completely separate system located outside the aircraft. It is driven by measured flight control surface positions which are hydraulically powered and fully operational during ground simulation. The onboard systems function identically in flight and on the ground.

The pilots are supplied with complete flight test displays on a CRT positioned outside of the windshield or the color flat panel display mounted on the instrument panel. The display presents a full three dimensional scene, including the runway environment, programmable tracking tasks, and project-specific symbology such as "pathway-in-the-sky" type guidance. Complete flight tests can be rehearsed on the ground using the same hardware, software, and pilot tasks which will be flown. This allows for very efficient use of the limited flight time available for low cost flight programs.

Flight Test Calibration and Data Recording

To aid the process of inflight checkout and calibration, the system is capable of fully automatic computer-flown test input sequences. A general purpose autopilot is implemented in the model-following system and coupled to a test input generator. This system can perform complex combination maneuvers such as holding wings-level, zero-sideslip flight while exciting the short-period mode with a sum of sines input and holding the average vertical flight path angle to zero.

A primary method of simulation verification is to use time history overlays which compare model responses to those of the inflight simulator. These can be produced in real time and displayed at the engineer's station. The model responses can be produced in advance or a separate explicit model can be programmed in one of the DSPs to run in parallel with the rest of the simulation.

Another tool under development is the use of onboard real-time Parameter Identification (PID). A computer generated sum-of-sines is flown by the autopilot to excite the aircraft at the desired frequencies. A Calspan developed frequency domain real-time identification technique has been implemented in one of the DSPs which provides linear state-space models containing the stability and control derivatives of interest.

The derivatives are displayed to the flight test engineer for review and may then be used to automatically update the model-following system gains. This allows specific test points in the overall flight envelope to be examined and calibrated with greater precision and ease.

The data recording system is fully configurable and is typically set up to save 64 channels at 10 millisecond intervals. The recording system is non-intrusive to the simulation because of the dual-ported RAM. The data are stored on the 90 Megabyte Bernoulli cartridges in Matlab format or any custom format required by the user. Because the media is random access, the flight data are immediately available for analysis or postflight briefings.

Video recording of the evaluation pilot's outside visual scene is accomplished using a miniature color camera (1/2" diameter by 3" length) located six inches outboard of his right eye. The high-resolution (410 line) NTSC signal is recorded on an 8mm two-hour cassette. Each frame is digitally encoded with the time and date for synchronization with the other flight data.

Inflight Simulation

Once the simulation is fully checked out, up to three flights (6 hours test time) per day may be flown. Typically, a four person flight crew is used with a Calspan safety pilot and flight test engineer, and a customer evaluation pilot with one extra pilot, engineer, or observer.

The simulation envelope allows for up to 325 KIAS and 2.8 g's. Altitude and velocity mismatches can be programmed and displayed to the evaluation pilot if desired. Safety pilot and computer monitoring of the simulation system allow safe landings to touchdown, formation flying, and simulated inflight refueling. Low L/D approaches at up to -17 degrees flight path angle can be accomplished with spoilers and landing gear extended.

Future Upgrades

Numerous future upgrades to the Calspan Learjet 25 are being considered and will be implemented based upon customer requirements. These upgrades include the following:

- a. Additional displays will be installed to add head-up, head-down, and helmet-mounted display capability. Custom formats and tracking tasks will be available.
- b. Servoed throttles and servoed flaps will be incorporated to provide control of additional degrees of freedom.
- c. The Learjet's multi-sensor navigation system will be interfaced to the VSS to provide space positioning data. This will allow the aircraft to evaluate flight control system designs which couple GPS or other navigation data into the flight control system.

Conclusion

A new inflight simulation tool is now available to the designers of aircraft and flight control systems. Thus far it has been used in the development of the Indonesian IPTN N250 regional transport, the Cessna Citation X business jet, and in several other flight research programs. State-of-the-art computer equipment and techniques have been and will continue to be developed which will allow engineers the opportunity to significantly reduce risk and produce a higher quality product while staying within their budget and time constraints.

ADDENDUM

THE LIMITS OF HUMAN IMPACT
ACCELERATION TOLERANCE

G.P.Stupakov*, Yu.V.Mazurin**

Russian Federation Air Force Institute
of Aerospace Medicine, Moscow

Abstract

The paper discusses experimental and theoretical aspects of grounding of new calculated criteria of trauma-safety and subjectively estimated by man the effect of impact acceleration exposure, considering the factors of pulse time duration enhancement and posture change, which are characteristic for modern ejection seats in high maneuverable aircrafts.

Nomenclature

a_i	accelerations in directions of co-ordinate axes with numbers i ($i=1-OX$; $i=2-OY$; $i=3-OZ$)
A	age
A_i	constant of approximation of empirical relationship of acceleration tolerance on axis i as a function of time
a_i^*	maximum admissible on tolerance criterion the acceleration of standard exposure on axis i
d	voluminous content of mineral substances in bone tissue g/sm
e_i	orths of related to seats co-ordinate system
F	relative diminution of spinal column segment durability
G_x, G_y, G_z	accelerations on axes OX, OY, OZ
HL	personal parameter of human
K_1, K_2	the number of cycles of applied loadings for comparable parameters
M	reduced torso mass, kg
R_i	dimensionless dynamic reactions of model (R_x, R_y, R_z) on corresponding axes
R_x^*, R_y^*, R_z^*	the values of dynamics reactions, corresponding to set probability of lesion
t_m	time
t_m^*	time of maneuver
V_i	velocity components of seat movement, m/s
V_i^*	velocity components, corresponding to maximum admissible accelerations of standard exposures on axes i , m/s
α	index of intensity of acceleration effect on restorative processes in bone tissue
γ_i	constant of approximation
$\delta_i, \delta_j, \delta_k$	acceleration, velocity and properly the relative displacement of reduced torso mass in direction of axis with number i m/s ² and m/s and m correspondingly
ϵ	deviations from standard pre-ejection readiness posture
τ	damping quotient
Ω_{hi}	time of acceleration pulse
	proper frequency of oscillations on axis i

The implementation of highly maneuverable aircrafts with thrust vector control system, equipped with ejection seats, which provide the adaptive regimes of their rocket boosters, necessitates the consideration of new factors, influencing on traumatic lesions of flyer and subjectively estimated by him impact acceleration tolerance. These factors are:

1. The decrease of mean value of amplitude of minimally traumatizing backbone of impact pulse longitudinal component due to lessening of spinal segment strength under prolongation of its effect, and bending and compression of spine frame against the background of high sustained $+G_z^*$ accelerations, embarrassing to flyer to take preparedness posture for ejection.

2. The modification of durability qualities of bone tissue under repeated exposure of impact loadings.

3. The decrease of subjectively estimated admissible limit tolerance of impact accelerations as time duration of impact impulses increases.

Let us consider consequently above mentioned issues.

For modern ejection seats and impact-absorbing devices it is very characteristic the augmentation of time duration of accelerational longitudinal component up to 2-3 s. The experimental data have shown the logarithmic dependence of mean value of minimally damaging the spinal column of ejection seat acceleration or equivalent overload $G_z^* = -a_z^*/g$ ($g = 9.81 \text{ m/s}^2$) on time duration of its action τ for interval of durations $0.06 < \tau < 3 \text{ s}$.

$$G_z^* = 18.413 - 3.122 \lg \tau \quad (1)$$

Changing on base of these data the durability parameters' distribution of most injurable segment of spine it is possible to determine for each fixed moment of time the value of admissible impact acceleration of catapulted seat, which corresponds to demanded probability of lesion.

The next factor, which should be taken into account at estimation of trauma-dangerousness of longitudinal component of impact acceleration vector, is the posture deviation, caused by as longitudinal as well as lateral and transverse components of vector of acting acceleration. It has been experimentally established, that during exposure of high-maneuvered and sustained flying accelerations a_{xm}, a_{ym}, a_{zm} the pilot practically lacks the possibility to take optimal preparedness posture, even in case of forced pulling to the ejection seat. The non-compensated by forced seat's restrained and fixation system residual bending of spine in sagittal plane in region of most traumatized segments of column $T_{12}-L_1$ increases as $-a_{zm}$

*Doctor of Science (Medicine), Professor

**Candidate of Science (Mechanical Engineering), Senior Scientific officer

heightens. This in turn leads to the more relative lowering of carrying capabilities of segment F in comparison with absence of additional bending in full accordance with relation, cited on Figure 1.

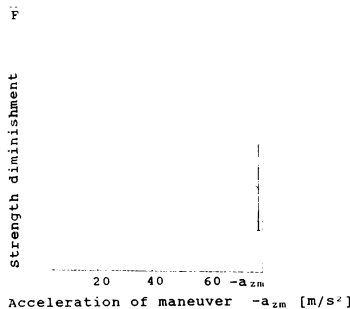


Fig. 1 The segment strength decrement at residual bending, caused by maneuverable accelerations.

The acting along spinal column axis the longitudinal component $-a_{zm}$ besides the residual bending provokes also the compression of spine. This circumstance on a level with decrease of admissible exposure rate by preliminary quasi-static for spine loading due to maneuvered aircraft flight accelerations, brings to non-linearity of strength characteristics of spinal column [1]. For example, $a_{zm} = -30$ m/s² increases the lowest frequency approximately 2 times.

The total effect of longitudinal impact pulse duration time, the residual bending of spine in lumbar region after putting in operation of forced pulling of flyer's body by restrained and seat fixation system and augmentation of its resonant frequency during compression by admissible amplitude of pulse of triangular form is presented in Figure 2.

The impact acceleration of braking during maneuver $-a_{zm}$ in combination with a_{zm} also may lead to additional completely non-compensated bending in lumbar region of spine, which lowers its carrying capabilities in axial loading.

The combined prolonged effect of $-a_{zm}$ and lateral component a_{ym} has produced non-stationary, progressing with time posture's modification, which is characterized by lateral bending of neck and lumbar regions of spine, not counting above noted bending of spine's lumbar region in sagittal plane. Inasmuch as existing systems of fixation and forced restraining are not oriented to the prevention of lateral displacements of flyer's body, the named above factor should be taken into account at the selection of bail-out regimen in conditions of prolonged aerial maneuvers with longitudinal and lateral constituents of flight energetic accelerations. In fi-

gure 3 there has been presented the dynamics of lateral displacements of head and pelvis under influence of $G_{zm} = 5$ and lateral flight accelerations G_y .

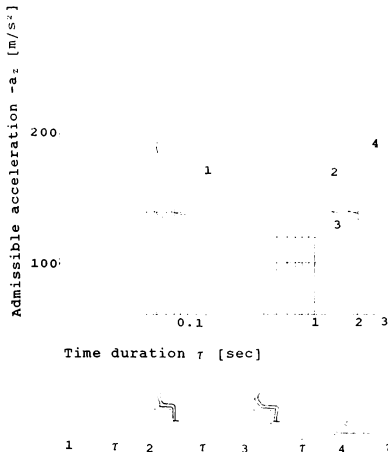


Fig. 2. Admissible level of accelerations a_z in dependence on duration and conditions of exposure:

- 1 - with consideration of strength decrement
- 2 - with consideration of (1), under effect of a_{zm} and at normal readiness posture
- 3 - in conditions of (2) and disturbed due to residual segment bending posture
- 4 - traditional (known) relation [2].

The formalized criterion of impact acceleration trauma-safety vector, which is voluntarily oriented relative to human body with consideration of first factor acquires view:

$$\left(\frac{R_x}{R_x'} + \left(\frac{R_y}{R_y'} \right)^2 + \left(\frac{R_z}{R_z'} \left(\tau, R_x(t_m), R_y(t_m), R_z(t_m) \right) \right)^2 \right)^2 = 1. (2)$$

where R_x, R_y, R_z - are components of dimensionless dynamic reaction in displacement of spatial one-mass model [2], described by set of 3 linear differential equations of second order

$$\ddot{\delta}_i + 2\epsilon_i \Omega_{ni} \dot{\delta}_i + \Omega_{ni}^2 \delta_i = a(t). (3)$$

For brevity of description in equation [3] the directions of axes OX, OY, OZ , connected with seat system of coordinates (see Figure 4), are denominated by orthonormal bases e_i ($i = 1, 2, 3$). The variables $\delta_i, \dot{\delta}_i, \ddot{\delta}_i$ denote the accelerations, velocities and relative deviations of reduced torsion mass M_i and $R_i = (M_i \delta_i / g)$ - are components of dimensionless reactions of models in corresponding directions.

**MULTIMEDIA DIGITAL DATABASE
INNOVATIONS AND APPLICATION TO
THE ANALYSIS OF WEAPON EFFECTS AND PERFORMANCE**

**James E. Welter, Capt, USAF
Defense Nuclear Agency
Structural Dynamics Division**

**Geoffrey S. Butler*
Horizons Technology, Inc.
San Diego, Ca.**

Abstract

This paper describes the development and application of software for the archival and retrieval of textual, audiovisual, and other types of data on the effects and performance of conventional weapons. A multimedia database management system, called the Weapons Effects And Performance Data Archival (WEAPDA) Information System, was developed to archive and retrieve comprehensive information covering over 30,000 operational uses of conventional weapons on various target types. This information will support various analyses including delivery accuracy, weapon effectiveness, weapon system reliability, and target vulnerability.

Introduction

The impact of low cost, personal computer (PC) based multimedia computer systems and applications are just now beginning to be felt in the commercial and government markets. Their capability to display textual

information, coupled with detailed graphics, still photographs and recently even full-motion video, has naturally led to applications involving education and training. These initial applications have proven the viability and cost-effectiveness of these systems, yet to date only a fraction of the potential applications for this technology are available or under development.

The recent introduction of compact disks (CDs) for data storage has further extended the range of applications for PC-based tools. With read only memory (ROM) storage capacity of 600 megabytes, educational materials, training manuals, and other cumbersome printed material consisting of thousands of pages of text and illustrations can now be placed on a single disk. In addition, the digital format combined with an estimated useful life of 25+ years, means that the information stored on CDs can be recovered flawlessly for decades.

Many uses exist for multimedia systems employing CD-ROM storage. This paper addresses the development of one

*Senior Member, AIAA

Copyright © 1993 by Horizons Technology, Inc. Published by the American Institute of Aeronautics and Astronautics, Inc. with permission.

such application by the Defense Nuclear Agency, which is designed to archive and rapidly retrieve data of all types (textual, photographic, video and others) on the performance and effects of conventional weapons used in the Persian Gulf War.

System Requirements

The system concept called for operation as a personal workstation capable of storing and rapidly accessing large amounts data in various media (textual, photographic, full-motion video, etc.). Initially these data were to provide as thorough coverage as possible of the operational use of conventional weapons during the DESERT STORM conflict. The ensuing database would then be used to support a variety of analyses of weapon system performance as well as the development of a knowledge base for munitions effects assessment. To realize this concept, data storage, hardware and software requirements were generated, as discussed in the following sections.

Data Storage Requirements

An existing textual database, developed by the Air Combat Command (ACC) and the Gulf War Air Power Survey office under dBASE®, was identified as the initial source database. This database consisted of 30,599 records covering the air-to-surface missions flown during the DESERT STORM conflict. In addition to these data, WEAPDA was to archive and retrieve thousands of pre- and post-attack still photographs of the targets which were attacked, over 5000 full-motion videos and accompanying audio of the weapons being delivered on the target, as well as interviews with seven

munitions effects assessment exploitation team members.

These interviews were highly valuable for their information content. The exploitation team members assessed the weapons effectiveness on-site, just after the cessation of hostilities, and provided expert opinions of the weapons effects.

Although these specific data requirements were known, the system was to remain flexible in order to accept other types of data (structural diagrams, maps, etc.) and data from other sources (other conflicts, test programs, etc.). Because of the large amounts of data already identified, and the fact that more data was likely to require storage at a later date, the database behind WEAPDA was also to be designed for enhanced efficiency. Also due to the large amount of information to be stored, WEAPDA was to maximize its use of digital compression technology in order to minimize storage requirements.

Hardware Requirements

The system was required to operate using off-the-shelf hardware. No system unique hardware was to be developed or applied.

Software Requirements

The software was to provide a flexible, intuitive user interface enabling a researcher or analyst to locate, sort, and extract desired information. Because users with varying backgrounds and interests were expected to use the system, data retrieval was to be straightforward, and yet provide users with

varying avenues for the retrieval of desired information.

The software also had to provide on-demand display of color still photographs and full-motion (30 frames per second) color video with audio. The capability to add, modify or delete data, in these varying media, was also required. The system architecture was also to remain flexible and allow for future growth.

System Development

Version 1.0 of the WEAPDA system was developed under a rapid prototyping effort and was fielded for operational testing eight months after contract award. Several challenges were encountered and overcome during the system development, as discussed below.

Database Design

As noted earlier, an existing dBASE® textual database was selected to form the core of the WEAPDA database. However, WEAPDA was to archive and retrieve related audiovisual information in addition to the textual description of the event. These data, especially overhead photographs, were pertinent to several entries in the textual database. For example, a single overhead photograph of an Iraqi airfield, such as the example shown in Figure 1, might show several aircraft shelters, each of which is the target within a separate entry in the textual database. Therefore, had a "flat" database structure been used, the photograph would have been unnecessarily repeated as part of each record.

To avoid this inefficiency, a relational database structure was developed and employed. Translators were then developed to read the information from the dBASE® database and insert it into the WEAPDA relational structure. The resulting textual database "core" was then reviewed for accuracy and completeness through direct comparisons with the original dBASE® database.



Figure 1. A single target is often the subject of multiple records (missions) within the database.

Software Development

One key goal was to make the system operation highly intuitive, so that very little training would be required to prepare an operator to effectively use the system. To meet this goal, the software development proceeded in several phases.

First, significant effort was expended establishing the system conceptual design. Functional block diagrams, such as the one for the database editing function shown in Figure 2, were used extensively to ensure that the development team had a complete understanding of the system functions and interrelationships.

Then a number of brainstorming sessions were held to design the flow of operation of the software, as well as prepare sketches of preliminary user interface screens. By design, these sessions included individuals with a broad range of perspectives. Members included software engineers, experts in audiovisual data archival, weapon effects specialists and others.

With this task completed, the preliminary user interface screens were actually coded up and linked together, so that the flow of system operations could be demonstrated. This "skeletal" system was then critiqued internally by a broad panel of software users and developers. The results of this review were then used to fine tune the system operations.

At this point software was written to support key system operations such as searching the database and retrieving and displaying archived information. With key functions in place, a preliminary design review was then held for members of the user community. Extensive notes were taken, and recommended system refinements were established.

This entire process, including two reviews of the system operations, was completed before the software coding was initiated. This streamlined the software development by minimizing changes once the coding was begun and also providing the development team a broadened understanding of the system as a whole so that code was developed right the first time.

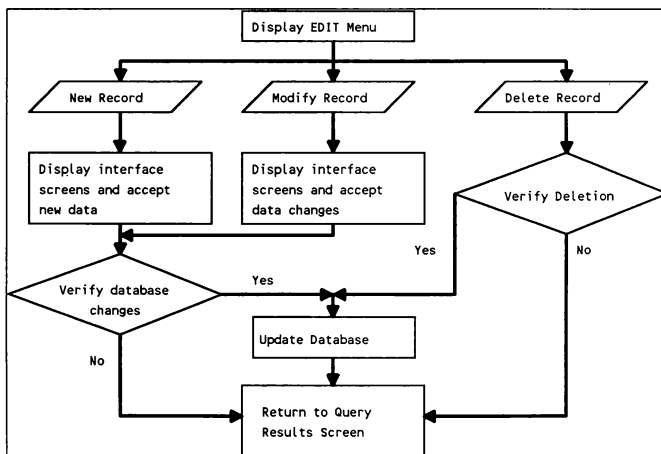


Figure 2. Sample conceptual design functional block diagram

Data Archival and Display

WEAPDA is a true multimedia database management system allowing the user to access and review textual, visual and audio information including maps, technical drawings, full-motion video and pre- and post strike high quality color photographs.

Textual data is stored in an efficient relational database which can be rapidly searched, reviewed and edited.

Still images are stored digitally in a number of formats, depending upon the desired image resolution. When image display is requested, the system recognizes the storage format and applies the appropriate decompression and display algorithms. The image can then be modified by adding text and/or symbols to highlight key features. Image enhancement functions, such as zoom, negative image, rotation and color modification, are also available to assist the operator in clarifying features.

The enabling technology for full-motion video display is Digital Video Interactive (DVI®), developed by Intel, which uses their i-750® chipset. This approach was selected principally for its good image resolution, symmetric (real time) capture capabilities, and ability to support other higher resolution formats¹. In addition, the i-750® enables the highest quality video playback under Indeo®, which provides a scalable, cross-platform video playback architecture.

Due to its data intensive nature, video is being stored on Compact Disk-Read Only Memory (CD-ROM). Over 250

CD-ROMs have been produced in the initial phases of the project.

WEAPDA Version 1.0 stores data related to the use of conventional weapons during the DESERT STORM air campaign, with emphasis on hardened and shallow-buried targets. This includes gun camera video of the actual weapon delivery during missions flown by the F-117, F-111 and F-15 aircraft, from weapon release through target impact. This is supplemented by satellite imagery and ground based still photography providing the pre- and post-attack data required to assess the effectiveness of the attack. Also included are interviews of exploitation team members providing expert assessments of weapon performance for selected key targets.

The system operates under Microsoft Windows™ 3.1, and takes full advantage of graphical user interfaces. Nearly all of the functions of WEAPDA are menu driven requiring a minimum of keystrokes, as illustrated by the Search menu shown in Figure 3. Textual data, such as the mission data shown in Figure 4, are presented upon request in individual windows, based upon data type, to aid comprehension.

The system is designed to help the novice user successfully retrieve and use the data contained. One of the features addressing this concern is a full-color, scrollable map function (Figure 5), which automatically tracks the mouse location on the digital map, and returns a requested search location. A help utility is available at all times to help the user with system operations, provide guidance, and define uncommon nomenclature.

WEAPDA Query Results									
File	Edit	Search	View	Options	Help				
BE #	Target	mc	Category	Lat	Long	Status	W		
0000-08016	Weapon		00051	332122	403549	MK			
00CA0469 2	Mission	H COMPLEX	07161	341441	410359	MK			
00CA0468 2	Data		07161	340108	403625	MK			
00CA0465 2	Damage Mechanism	IPLEX	07162	331147	411618	MK			
00CA0463 2			07162	325931	405645	CO			
00CA0543 2	Sort		07600	333050	425626	MK			
00CA0543 2	Find		07600	333050	425626	MK			
0000DA0017	AL QURNAH BR W OV NAHR AL FURA		04110	310019	472510	MK			
0000DA0017	AL QURNAH BR W OV NAHR AL FURA		04110	310019	472510	MK			
00-08006 2	H 3		00051	325551	394449	CO			
00-08023 2	AL TAQADDU		00040	332024	433558	CO			
0000-08813	AL SABER SSHELTER		00052	285611	474729	CO			
0000-08813	AL SABER SSHELTER		00052	285611	474729	GB			
0000-08813	AL SABER SSHELTER		00052	285611	474729	CO			
0000-08814	ALI AL SAUR BASE		00051	292053	473115	CO			
0000-08814	ALI AL SAUR BASE		00051	292053	473115	CO			
0000-08814	ALI AL SAUR BASE		00051	292053	473115	CO			
0000-08814	ALI AL SAUR BASE		00051	292053	473115	CO			
00TL0105 2	SA2 AL ABR		07220	292130	470720	CO			
0000TT0191	KILL BOX A		00210	291500	464500	AG			

Figure 3. Search requests are prepared using simple pull-down menus

Mission Data			
Installation Summary			
BE#:	Name:	Country:	
0000-08813	AL SABER SSHELTER	Kuwait	
ATO:	Date: 01/18/1991	Tot. Mission Aircraft: 2?	
Mission #:	0711F	Aircraft on Target: F-16	
Unit:	388TFW	Quantity: 1	
Callsign:	CHANKER 13 ?	TOT Start Time: 01:07	
Type:	OCA	TOT End Time: 00:00	
Linesort:		Directed By:	
Source:	MSR	Weather on Target <input type="checkbox"/> Mission Cancelled	
<input type="button" value="Print"/>		LIGHT CLOUDS AT 12,000 FT.	
<input type="button" value="Visual"/> <input type="button" value="Target"/> <input type="button" value="Weapon"/> <input type="button" value="Reports"/>		<input type="button" value="X Cancel"/> <input type="button" value="? Help"/>	

Figure 4. WEAPDA mission data summary screen.

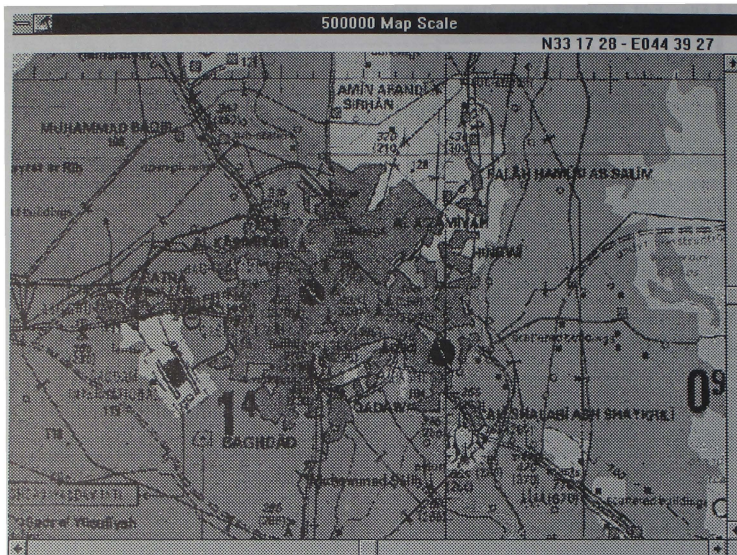


Figure 5. Three scales of maps are available to assist the user with target location.

Hardware Selection

The WEAPDA software is hosted on an upgraded stand-alone 486 desktop PC operating at 33 Mhz. Included with the basic PC are a 150KB/sec and a 300 KB/sec CD-ROM drive, 800 MB removable hard drive, a stereo speaker set, and 16 MB of random access memory (RAM). Fielded systems have been equipped with a 14 inch SVGA monitor, although future stations may opt for larger screen sizes.

Applications

Potential applications for multimedia database systems such as WEAPDA

appear unbounded. The WEAPDA database in particular will have application to a wide range of user requirements.

To a system analyst, for example, investigating the effectiveness of a weapon system or vulnerability of a target to various weaponizing strategies, the WEAPDA system places available historical data as well as a powerful analysis tool at his fingertips.

For the air campaign planning staff officer charged with reviewing, analyzing, and refining a target list prior to development of the Air Tasking Order (ATO), WEAPDA could be a source of

textual and visual data on specific target or historical data relating to "similar" targets. The availability of such information will help the staff officer "visualize" the target.

To the weaponeer, utilizing a weaponeering decision support system, such as DNA's Munitions Effects Assessment (MEA) tool, WEAPDA will provide a source of data rich case studies for various target types. Making use of expert opinion and intelligent technologies, knowledge and experience of these case studies can be applied to "similar" targets for which intelligence data might be limited or non-existent.

To the research engineer, WEAPDA can provide a convenient multimedia database management system for analyzing, archiving and retrieving valuable test data. With networking, various laboratories can share and exchange textual and visual test data results.

To the historian, WEAPDA provides a state-of-the-art historical reference system. By performing simple database queries, the historian can access and review both textual and visual data on a particular event, campaign, weapon system, or test at a single desktop workstation.

Conclusions and Future Work

The WEAPDA system software database development program will continue with expansion and improvements to the system software and database.

The system shall be expanded to include effects and performance data from past

and future conflicts, inclusion of test program data, possible inclusion of hard target imagery from countries of interest, and further refinement and population of the existing DESERT STORM database.

The cost of developing and maintaining stand alone WEAPDA systems (cost of configuration management and replication of multiple sets of CD-ROMs) becomes a concern once the system is replicated beyond the initial three beta test systems. A promising alternative to stand alone systems is to develop a Wide Area Network (WAN) and serve the data to all potential users from a central server location.

Because the data is classified SECRET-NOFORN, the data must be encrypted before transmittal. This problem can be simplified by making use of recently developed technology which will encrypt the data before it is written to CD-ROM. This capability has been approved by the National Security Agency (NSA), and is available today. The network administrator can then operate the system in an unclassified, open environment. The CD-ROMs are permanently keyed, so there is no security maintenance required.

Additional plans for WEAPDA include porting to platforms operating under UNIX. This would allow a host of new software systems currently under development to access, query, and import multimedia effects and performance data directly from WEAPDA.

References

- 1) Walker, P.E. and Philipot, M.,
"Off-line Time Asymmetric Compression
Techniques for High Quality Digital
Video," First International NAB/IMA
Multimedia World Conference
Proceedings, April 1993.

COMPUTERIZED TEACHING OF PILOTS
TO SPATIAL ORIENTATION FLIGHT TASKS

V.A.Ponomarenko*, A.A.Vorona**, D.V.Gander, V.M.Usov

Russian Federation Air Force Institute
of Aerospace Medicine, Moscow

Abstract

In this report there are stated new methods and psychophysiological technologies of formation of mental actions and flight image as one of phases of psychophysical preparation. There is presented the innovative computer program of stabilizing training to spatial desorientation.

Man's ability to adequate perception of his piloted aircraft spatial position, supporting situational awareness of its spatial orientation and motion relatively Earth's surface at every instant of time on a basis of visual flight indicators information and encoding it into a spatial image is an important feature of the human factor in the control system of modern highly maneuverable aircraft. However this ability is not inherent one and it should be purportedly trained in professional pilots by their enclosure of special tasks and exercises into instructional system, training and attesting pilots flying skill at the initial stages of their flight preparation and subsequent retraining for transition to new types of aircraft. In contrast to habitual mode of man's movement on land surface for spatial orientation in flight the pilot cannot be satisfied by ontogenetically formed reflectory reactions, he needs to have complicated visual-imaginary notions which step out as a basis for formation of new sensory perceptive patterns in the conditions of conflicting accelerations influence.

For effectively carrying out mental actions over interpretation of flight instrument and non-instrument information to aware aircraft position on flight trajectory, as it has been shown in several psychological and pedagogical works, it is necessary to develop new technologies of teaching and training for formation of psychical features and qualities, which will provide psychological readiness to activity, and also active and directive formation of mechanisms of operational thinking that must be actualized in complicated flight situation [1-3]. In more general statement the task of formation of pilot psychical readiness to control of highly maneuverable aircraft, to our view, should be considered from the position of flight image conception [3], in the light of which the major task of development of pilots teaching and training methodology is the formation of regulating mechanisms of pilot activity and some professionally important qualities, connected with integrating intellectual functions of psychical activity, in particular, with cogniti-

ve mechanismus of activity regulation. This opinion is based of the following, known out of literature, factors [4, 5]:

- purposeful choice of methodical approaches and technical means of instruction allow to form quite separately conceptual and executive mechanisms of activity;

- possibility of man adaptation to fast and sudden changing conditions of performance is to a considerable extent provided by development of cognitive mechanisms.

Let's cite psychological content of teaching course which is realized in the presented computerized teaching system and which expands analogical requirements to traditional means of pilots ground training [2, 4]:

- formation of notions and knowledge of spatio-temporal characteristics of acrobatic maneuvers;

- development of orientation base of decision making process under characteristic errors in flying technique;

- training of spatio-temporal perceptions of aircraft movement on the trajectory;

- training of distribution and switching attention on indicators, formation of images of display indications in the prominent cue points of acrobatic maneuvers;

- development of anticipated mental actions;

- formation of notions and knowledge about informational signs and temporal relations of their appearance in special cases of flight;

- formation of tactical thinking as a basis of making decision about optimal maneuver choice under different situation parameters and at various stages of flight;

- formation of psychical readiness to actions of recovery of aircraft out of complicated position under different situation parameters at various stages of flight.

The essential result, expected from computer-aided teaching of pilots, is provision of the flying training methodist with possibility of qualitative solution of pedagogical and psychological tasks in which the requirements to development of profession-important ability of performing spatial orientation and reencoding instrument informational image into aircraft spatial position image are the most prominent.

*Academician, Doctor of Medicine

**Candidate of Medical Science

Candidate of Psychological Science

Candidate of Medical Science

In contrast to the earlier used traditional methods of flying training on the ground facilities the computerized system was created with the aim to provide:

- intergrity of perception of acrobatic maneuvers;
- formation of exact spatio-temporal notions about aircraft trajectory movement;
- visual demonstration of cause and effect connections in the chain "error - aircraft trajectory deviation - a real code of flight instrument indications".

Under creation of methods and scenery for computerized teaching system there were taken into consideration the above-mentioned psychologo-pedagogical requirements, the content of which is presented in the following scenery peculiarities:

- 1) Over all acrobatic maneuvers there is foreseen building of the correct (standard) trajectory with corresponding parameters in the prominent cue points and presented possibility of changing image scale and attitude in dependence on "observer" position. This allows to realize principle of obtaining "view outside in" in process of teaching.
- 2) For each acrobatic complicated maneuver there are formed spatio-temporal notions, realistic over dynamics of trajectory movements in the real time scale and under retaining relative spatial volumes of flight maneuvers. This allows to realize the principle based on the pilot true time perception and provide imparting habits of realized control of time function to the pilot.
- 3) There is foreseen possibility of visualization of task consequences in the form of concrete parameters meanings out of any prominent cue point of each of complicated acrobatic maneuvers, that provides demonstration of cause and effect connections in the chain of events "decision over control-trajectory position - changing of pilotage parameters".
- 4) Compiled exercises collection supposes gradual complication of the training task with support of earlier learned materials. So, if in the first exercise the trainee simply gets to know spatio-temporal maneuver trajectory and in the second exercise he learns the major meanings of its parameters, then beginning with the third exercise there takes place control of parameters knowledge in the trajectory prominent cue points, in the fourth and the fifth exercises the trainee gets acquainted with typical errors and their consequences. On the final stage he learns the order of actions during error corrections, and selfcontrol of the function of spatial perception of an acrobatic maneuver. This allows to provide formation of psychical processes of foreseeing and include elements of selfcorrection of learning process.

Thus, the computerized teaching system, taking into consideration the requirements of general didactic principles, as visual aids, gradual increase of learning material complication, adaptivity and

problematical character of learning, realizes also such principle of psychological training as the principle of priority of mental actions and trainings.

Expert-consulting aspect of computer teaching system application is connected with the fact that wide spread of personal computer systems gave rise to a new class of teaching programs - the programs supporting computational experiment in the course of which a trainee can model a complicated system performance in some initial conditions, for example, carrying out complicated acrobatic maneuver under such modes and at such altitudes, which in principle is impossible to reach in real conditions without risk of losing a man and an aircraft.

Equipment limitations knowledge is also that essential feature of actively learned knowledge, which permits consciously to run risk in extreme unexpected cases, requiring to make a special decision.

Thus, we can formulate the following base position of this report: the development of modern informational technologies allows, using means of computerized training, to solve the tasks of formation of professionally important pilot's qualities and among them ability to perform spatial orientation. In this context computerized teaching is realized as one of technological directions of flight modelling, the essential features of which are:

- repeated presentation of flight path during performance of complicated acrobatic maneuvers in the form of trajectory track (in coordinates of external observer-out in) with visual demonstration of results and consequences of any pilot's possible decisions about the way of performing maneuver and recovery an aircraft out of complicated position in case of early admitted error action;
- computerized synthesis of aircraft control process, as a mean for "ideal" maneuver (task) performance or demonstration of embedded in error decision making action methodology;
- computerized image synthesis with high degree similarity of spatio-temporal parameters during maneuver and high reality of flight display indications of piloted aircraft flight track and its attitude (projection on display screen at any voluntary external observer position selection);
- storage in computer memory of all intermediate results of modelling for repeated presentation to instructed subject at any temporal intervals and in the most prominent cue points on the flight trajectory according to the determined didactic task of learning program;
- storage in data base of the results of testing of subject abilities to perform spatial orientation task with obtaining quantified time and error decision making probabilities on a base of experimental observation sample.

Each of the above mentioned positions has psychophysiological substantiation from psychological concept of "flight

image", which has been developed under the leadership of the first author of this report.

Real time realization of three-dimensional computer-generated graphic images on display screen is a very important thing for formation of adequate temporal perception of consequently occurring events ("time sense", which may be significantly distorted in real flight situation, provided, that the pilot is not sufficiently trained to flight mission), involuntary "switching on" in the trainee the reaction of anticipation, mental "foreseeing" of aircraft flight path (as ability to dynamic spatial observation) and its correction based on the results of observation for flying trajectory, activation of cognitive psychic processes during selection of the best position for external observer (because this provides selection of voluntary projection and voluntary distance of point of observation) and consequences of introducing by a trainee himself typical errors of aircraft maneuver performance.

The synthetic visual generation of the piloted vehicle on the trajectory as a three-dimensional object with constantly changing attitude positions and simultaneous presentation of flight indicator code, fixation of orientational points of complicated acrobatic maneuvers on the trajectory constitute the important part of computerized learning methodology along with possibility of setting various initial conditions (simulation) and demonstration of different degree complexity exercises for introducing in flight training course elements of adaptivity to the individual program of pilot teaching. The principal setting during development of the computer system was significant narrowing of the trainee performing actions and on the contrary increased load on the intellectual sphere with actualization of base psychic processes and functions (memory, attention, thinking, perception). In all cases of development of new exercises special importance was attached to creation of adequate testing regimen. So, under studying prominent cue points on the trajectory during performing complicated acrobatic maneuvers the obligatory link was the test on knowledge of critical parameters values in each prominent cue point, and under mustering methods of timely detection of deviations from nominal trajectory of vehicle movement there was supposed at the final stage to perform testing of this ability with subsequent passing to regimen of studying methods of recovery vehicle out of this position. Under mastering some exercises there was attached great importance to automatic detection and warning the trainee of dangerous deviations of flight parameters values, that, in designers opinion of this learning system, assisted a pilot in developing circumspection and adequate notions about possibilities of vehicle and control system. Analogous control took place under setting initial conditions of modelling, because this assisted to master knowledge about "working" ranges of initial parameters under all typical maneuvers and their typical combinations according to flight manual.

The applied program support of computer teaching system written, using the

high level programming language, warrants "friendly interface" with user (trainee and flight instructor) in the conditions of the mass application of standard software and hardware-configuration apparatus of personal IBM compatible with IBM PC AT 286/386.

The work with computer teaching system does not suppose special education of the user in the field of informatics and computer equipment. The system provides by "hot key" presentation of the context-dependent help with full instruction over the actions, expected from operator and characteristics of the concrete learning regimen. Besides, on the screen there is constantly present information in the form of "help line" with indication of current "hot keys". In the case of operator error actions there appears different warning information, and in the general case the error actions are blocked by the system. Significant time reduction of choice of the task and system adjustment in learning regimen are reached by presenting many meanings of input parameters "over omission" and methods of control with the help of the mouse manipulator or the limited keys list (arrow, Enter, Esc and functions keys F1-F10).

Having chosen the indicators of timely pilot recognition of disturbances of spatial trajectory vehicle movement, as the main criteria in the spatial orientation task, and probability of error recognition of the complicated position and error decision making about methods of piloted aircraft recovery, the indicators of psychological man reliability in this situation, we have succeeded in showing that in contrast to control subjects, who did not receive computer-aided flight teaching, in the experimental subject group:

- the spatial orientation position determination time has decreased twice along the error diminishment by 10-14%;

- the pilot decision making time on subsequent aircraft recovery maneuver out of the complicated spatial orientation position (after intentional putting aircraft into it) has reduced by 2,5 times;

- the probability of error actions in the mentioned conditions has fallen from 0,86 to 0,11.

The received data allow to formulate some practical conclusions:

- first, it is highly expedient to use computer technologies for formation in aviation students complex "visual-Gestalt" notions which create the basis of spatial orientation skills for piloting high-maneuverable aircraft;

- second, it is necessary to provide special training sessions for developing in aviation students of professionally important qualities, increasing reliability of pilot actions during recovery of controlled aircraft out of complicated spatial orientation positions.

Principally new moments which distinguish the modern state of creation of the wide class of computer-aided teaching systems of pilots are:

- the success in the development of psychological theory on a basis of which there are created modern methods of flying teaching based on the regulation mechanisms of man behaviour and performance in the complicated system "pilot-aircraft";

- the general scientific public realization of the new state of affairs of informatics and its role in the development of many aspects of scientific-technical policy of professional education, training and additional training of personnel of modern complicated industry and transport;

- the appearance of computer equipment of personal computers type and new informational technologies, adequate to new methodical approaches.

At the present time these new means attract great attention of specialists due to the following distinctive features:

- they are not expensive in comparison with traditional simulator - modelling complexes, accepted for application in aviation;

- they are highly mobile "in the sense of adjustment" to the concrete teaching-pedagogical task and allow to realize "friendly" interface for the trainee of any level education and for the flight teaching methodist;

- they allow to be a peculiar training ground during development and approbation of different teaching methods at the expense of the newest informational technologies.

References

1. Beregovoy G.G., Lomov B.F., Ponomarenko V.A. Experimental psychological investigations in aviation and space. M., Science, 1978, 303 p.
2. Gurdzhian A.A. Spatial orientation of the pilot. In the book "Aviation Medicine" (Manual) edited by N.M.Rudny, P.V.Vasiljev, S.A.Gozulov M.: Medicine, 1986, P. 339-351.
3. Zavalova N.D., Lomov B.F., Ponomarenko V.A. The image in the system of psychological regulation of performance. M.: Science, 1986, P. 174.
4. Kartamishev P.V., Ignatovich M.V., Orkin A.I. The flying teaching methods. M.: Transport, 1987, P. 279.
5. Kovalenko P.A. Spatial orientation of pilots. Psychological features. M.: Transport, 1989, P. 230.

TEXTURE AS A VISUAL CUEING ELEMENT
IN COMPUTER IMAGE GENERATION
PART I - REPRESENTATION OF THE SEA SURFACE

Greg Bookout
Digital Visions
Mt. Shasta, California

John Sinacori*
Montallegro Applied Sciences, Inc
Pebble Beach, California

Abstract

The objective of this paper is to advance hypotheses about texture as a visual cueing medium in simulation and to provide guidelines for data base modelers in the use of computer image generator resources to provide effective visual cues for simulation purposes. The emphasis is on a texture decoration of the earth's surface data base in order to support low-level flight, i.e., flight at elevations above the surface of 500 feet or less. The appearance of the surface of the sea is the focus of this paper. The physics of the sea's appearance are discussed and guidelines are given for its representation for sea states from 0 (calm) to 5 (fresh breeze of 17-21 knots and six-foot waves, peak-to-trough). The view-points considered vary from 500 feet above the mean sea surface to an altitude just above the wave crests.

Definition of Texture

"Texture, as a component of appearance (of an image), is a spatial array of patches or spots with varying degrees of regularity in their size, shape, arrangement, distinctness, color and brightness. Texture gradients refer to systematic variations in the above factors over space. Texture shear is a discontinuity in the spatial patterns." The preceding definition is taken from Ref. 1. It is a broad definition that encompasses texture patterns ranging from checkerboards to tree crowns in a contiguous forest. It

could also refer to an array of highly-structured images such as a contiguous array of photographs of a familiar face. As such, one could argue that the definition encompasses all imagery, however, the term "varying degrees of regularity" is key to the description of texture because it describes repeating patterns. Texture can occur in both static and dynamic forms. The checkerboard and tree crowns (without wind buffeting) are examples of static texture, i.e., texture that varies only over space. A water surface moving under the combined action of wind and gravitational forces is an example of dynamic texture, because the patterns vary over space and time.

Visual Cues Afforded by Texture

A visual cue is an array of light that can be perceived by the human visual sense to be a prompt for action. As such, the concept ties together "perception" and "action" implying some principled basis for the relationship between them. A simple example would be the visual cue offered by a familiar object being approached for the action of braking in order to avoid striking it. The visual cue here is the apparent size of the object which can be perceived to be at a certain distance away. When the apparent size reaches a value corresponding to the distance in which braking must be initiated, the prompt for braking occurs. The principled basis is the inverse relationship between the apparent size of the object and its range, which can be derived using the principles of Euclidean geometry. Suppose that the object is a cube whose dimensions are not revealed to the observer. Now the object, although having a familiar shape that is readily perceivable, does not have a familiar size and a principled basis cannot be derived from geometrical principles. Given that the

*Member, AIAA
Copyright ©1993 by John B. Sinacori
Published by the AIAA, Inc. with
permission.

cube cannot be observed using other sources of range information such as from the use of two eyes (stereopsis and convergence) or accommodation (focus reflex), can the task of braking to avoid a collision with the cube still be accomplished? The answer is yes. The principled basis that supports the visual cue is dynamic, non-intuitive and can be improved by texture on the cube's faces.

The quality of a visual cue is imbedded in its principled basis. If perception of an object's apparent size is subject to error, then so is the range estimate. The relationship may be derived by taking differentials of range and apparent size. If the cube were now shifted laterally to one side so that it showed one of its sides, the errors associated with the principled basis could change, yielding a different precision in the determination of range. If a second cube of identical size and orientation were placed in the lateral conjugate position of the first cube (mirror image), would the precision in determining range improve? Probably not. A mirror image of a cube does not fundamentally provide any more information than does the cube itself. This point demonstrates that visual cues can be redundant.

The above points illustrate two properties of visual cues: 1) their quality of information conveyed, which is imbedded in the principled basis, and 2) their quantity. Both properties are important to the discussion of texture. An important property of texture is its degree of variability. This can be expressed many ways, however, a common one is to describe the ratio of the areal sizes of the largest to the smallest patch. Another way is to describe the ratio of the largest diameter to the smallest diameter of circles whose areas are equal to the areas of the corresponding largest and smallest patches (equivalent diameter ratio). When these ratios are equal to one, the texture pattern is like that of a contiguous checkerboard. When the ratio is very high, e.g. in the thousands, and the patches are distributed randomly, the regular appearance is lost and the texture appears constant no matter what the orientation or range of the observer. Consider a textured surface viewed through a window. If the variation in texture, size or equivalent diameter is significantly smaller than the variation in range to the surface, then texture gradients will be apparent and a cue of relative range to all

parts of the surface from the viewpoint is presented. Also, the cue of orientation of the window center line-of-sight relative to the surface normal is present. If the surface is undulating, i.e., not flat, a cue to the surface shape is present. If a discontinuity in range to the surface exists, e.g., a near hill against a distant one, texture shear will be present provided the variation in range at the discontinuity is larger than the texture variation.

Statically, therefore, i.e., no movement of the viewpoint relative to the surface, the visual cues potentially contained in texture are, in flying jargon, absolute pitch and roll attitude, relative range to all parts of the surface from the viewpoint, and relative surface shape. Heading change cues are also present under conditions of continuous viewing. The principled bases for these cues is the relationship between texture patch mean size and distance to that patch from the viewpoint. The quality of the range and shape cues is strongly affected by the texture variability relative to the range variability.

Dynamically, much more information is potentially available. Clearly, attitude rates (and higher derivatives) are potentially perceivable from cues of window, viewpoint and observer movement relative to the texture array. The principled basis here is the relationship between the window (and observer's) angular velocity vector and the apparent angular velocity of each patch of the texture array, sometimes referred to as "optical flow." This relationship is unique, if there is no relative movement among the texture patches themselves, i.e., a rigid surface.

Translational movement of the viewpoint and observer relative to the textured surface offers additional information about absolute range, direction of viewpoint motion relative to the surface, and absolute surface shape, regardless of the texture and range variations. The principled basis for this lies in the relationship between the apparent angular velocity of each texture patch and the velocity of observer rectilinear motion relative to the surface. The relationship is again unique, if the surface is rigid. This principled basis is motion perspective, as termed by Gibson (Ref. 2). It has been analyzed (Ref. 3) and found to describe the relationships among

the apparent angular velocity of every texture patch, aim point and time remaining before passage by that patch and, therefore, absolute pitch, roll and time remaining before passage by each patch and surface shape, provided the surface is rigid.

Therefore, it is reasonable to assume that pilots flying at low-level over random-appearing textured surfaces can perceive their state (attitude and position) and surface shape solely by the principled basis of motion perspective. This point was proven in Ref. 3, where the performance of an aircraft automatic terrain-following system was defined in which the system used only a camera pointed generally ahead and a texture flow-field processor. No image feature recognition algorithms were employed. A finite time sample of the imagery is required, however, in order to acquire the cues of texture patch apparent angular velocity. This point is important because it says that pilots must look for some finite time (1-2 seconds) in order to perceive their state, and this may have implications regarding visual attention and workload.

Furthermore, the analysis of Ref. 3 not only showed that motion perspective is sufficient for low-level flight, it also pointed out that the quality of this principled basis varied widely with the quantity of visual cues presented, i.e., the quantity of texture patches visible. Proof of pilots ability to fly at low-level using texture only is described in Ref. 4.

When texture patches themselves move relative to each other, another principled basis is suggested. Of course, this could refer to the textured appearance of the surface of the ocean. Not only does it exhibit the appearance of texture, but the texture patches themselves are dynamic, i.e., they change with time. Waves are generated by the combined action of wind and gravitational forces. Wave dimensions and time histories are predictable in statistical terms. A large wave moves differently than a small one. This property suggests a principled basis exists that is the relationship between a waves time history and its size. Therefore, it could be argued that a cue of absolute range to each visible wave is present. This means that cues of absolute pitch, roll and translational position are offered by the dynamic sea

surface under conditions of static viewing, i.e., the observer is fixed, as in a hovering helicopter.

To summarize, four statements and Table 1 are given below:

- 1) Visual cues yielding absolute pitch and roll attitude and relative range and surface shape are potentially available from a statically-textured surface under static viewing conditions.
- 2) Visual cues yielding absolute pitch and roll attitude and absolute translational position are potentially available when the observer moves relative to a statically-textured surface.
- 3) Visual cues yielding absolute pitch and roll attitude and absolute translational position are potentially available when the observer is fixed relative to a dynamically-textured surface.
- 4) It follows, logically, that under conditions of both dynamic texture and dynamic viewing, information about absolute pitch, roll and position relative to the textured surface is available through the cues provided by the texture.

TABLE 1. Flight Information Available from Texture Cues

(Upper case - absolute, lower case - relative;

P - pitch, R - roll, Ra - range, S - surface shape)

	Static viewing	Dynamic viewing
Static texture	P, R, ra, s	P, R, Ra, S
Dynamic texture	P, R, Ra, S	P, R, Ra, S

The last point to be emphasized concerning visual cues afforded by texture is the resulting accuracy of the perceived information and habituation or long-term effects on perception of position. Reference 4 cites analyses of data from three experiments on terrain-following tasks utilizing texture, linear perspective, and both. It was found that regulation of the height above ground was poorer with texture alone than with linear perspective, the height errors with texture alone being approximately twice those with linear perspective. Furthermore, the results with both were the same as with linear perspective, suggesting that the human operator preferred the more

accurate linear perspective cues.

Habituation, or long-term effects on perception, may occur with texture. The visual cues that can be perceived into height above the surface are encoded in the high angular velocities of surface features rushing by the aircraft just ahead and to the sides. With prolonged exposure to such a flow field, the impression of closeness to the surface may diminish, producing an inclination to fly lower. Also, it has been found by researchers on automobile safety that placing cross stripes on a roadway whose spacing decreases yields a decrease in driving speeds. Presumably, this effect is due to the increased rate of feature passage. Also, data from experiments reported by Warren in a personal communication suggest that the perception of closeness to a surface not only depends on the cues of feature angular velocity, but feature passage rate, which means the local spatial density of features. Pilots training for low-level flight are made aware of potentially dangerous situations where they must fly from a higher density feature area to one with a lower density. The pilots are warned of the possibility that they will perceive themselves as having climbed, therefore inducing an incorrect prompt to descend.

It is reasonable to assume that the use of texture by pilots is not the best visual strategy given the availability of better cues from linear perspective or other sources. In this regard, visual cues from vertical objects such as trees, poles and towers, roadways and stream beds may be visually preferable over texture, because the perceptions can be made quickly, as opposed to the use of texture, which requires time and therefore visual attention.

Texture Representation of Water Surface

Physics of the Sea's Appearance

The following is addressed to the appearance and representation of the sea surface for sea states from 0 (calm) to 5 (fresh breeze of 17-21 knots and six-foot waves, peak-to-trough). The viewpoints considered vary from 500 feet above the mean sea surface to an altitude just above the wave crests. The appearance of the sea surface primarily depends on four factors:

- 1) The wind, which disturbs the surface.
- 2) Gravitational attraction, which attempts to restore its equilibrium.
- 3) The reflective power of water, i.e., the ratio of the intensity of a reflected ray to the corresponding incident one, which is a function of incidence angle.
- 4) The luminous intensity of the sky.

The description that follows is ordered according to increasing wind speed and corresponding sea state. When the wind is calm, the sea surface resembles a mirror and sea state is zero. The sky and objects in it such as the sun and moon are imaged in the water. The appearance of the surface, however, is not like that resulting from a mirror, but more like that from a pane of thick glass without a reflective coating. The reflecting power of a typical glass surface is a function of ray incidence. The reflecting power is the ratio of the intensity of the reflected ray to that of the corresponding incident ray. It is 100 per cent at zero incidence and about 4 percent at 90 degrees incidence. If a water surface exhibits this characteristic, then the luminance of a calm sea will be equal to that for the sky near the horizon and the sea surface luminance at the nadir will be about 4 per cent that of the sky at the zenith. The luminance of a point on the surface between these values, therefore, is related to the luminance of its conjugate point in the sky by the surface reflecting power relationship, where incidence angle is equal to the depression angle of the line-of-sight (LOS) below the horizon. The conjugate sky point is at an elevation angle equal to the LOS depression angle. The color of the sea is a complex subject and well beyond the scope of this work. Some insights into the complexities are offered in Ref. 5, especially for shallow waters near shores. For a first approximation, it is reasonable to assume that the spectral properties of incident light are unaltered following reflection by the sea surface. This means that the color of a point on the surface is a reduced intensity hue of the conjugate point in the sky according to the relation previously described. This assumes that the absorbed portion of the incident light is not scattered back into the atmosphere, a condition that is mostly met in deep water.

The color of the sea near shores is greatly dependent on the nature of the bottom and is more fully discussed in Ref. 5.

As the wind speed increases, shear and normal forces distort the surface and the gravitational attraction with friction tends to restore it. The results are waves. At wind speeds of 1-3 knots, wave heights reach 1/4 ft., and the sea state is 1. A uniform wind in a constant direction would create a criss-cross pattern of intersecting waves moving nearly at right angles to the wind direction.

At wind speeds of 4-6 knots, waves reach a height of 1/2 ft. and the surface becomes more irregular. The sea state is 2. Wind speeds of 7-10 knots create waves up to 2 ft. high. These waves are not quite high enough to appreciably disturb the air flow over the surface, so the surface retains a glassy appearance punctuated by waves varying in height from a fraction of an inch to 2 ft. high. While the larger waves are created by wind and gravitational forces, the smaller "capillary" waves, about 1/2-inch high, are created by wind and surface tension forces.

As wind speeds reach 7-10 knots, the wind flow is disturbed by the surface and the nature of the sea surface becomes much more irregular. Reference 6 contains an excellent description of ocean waves: "Hence, for all wind speeds above 6 knots - if not before - we can no longer follow the growth of representative, nearly identical waves, but must adjust our sights to deal with a statistical ensemble (directional spectrum) of waves, moving within roughly 50° of the wind direction. The ensemble contains waves of all heights and periods from a lower limit dictated by capillary dissipation to some upper limit determined by a balance between the rates at which energy is supplied by the wind and removed by dissipative processes - principally breaking."

The preceding was mostly a description of the sea surface shape and the mechanisms that cause the shape. The appearance of the surface may still be assumed to result from the reflection/absorption relation previously discussed except, of course, for areas with crest-breaking foam and spray effects which begin to occur at sea states above 3. A single wave is visible

only because its face, facing the observer, is inclined, causing more incident light to be absorbed than at the neighboring surfaces. Also, a wave cannot be seen when viewing it in a direction along its crest, because the incidence angles on each side are similar, yielding similar reflection, and thus no contrast. We can conclude, therefore, that waves mostly broadside to the observer's LOS are visible, even though many other waves may exist in the region viewed that are not visible because their crests are nearly parallel to the observer's line-of-sight direction. This is why the sea surface at sea states from 1 to 5 appears independent of azimuth of the observer's LOS direction, except when large swells are present.

The same basic appearance of the sea is retained up to sea states of 5. This is an array of darker patches of varying sizes elongated parallel to the horizon, but with more or less the same basic shape. If these were viewed continuously, we would observe that the waves rise from the neighboring surface, grow to a maximum size, move a short distance and then fall back into the milieu of the surrounding surface and disappear. At the higher wind speeds, the brief movement is generally in the direction of the wind. At a sea state of 3 (windspeeds of 7-10 knots), large wave crests begin to break, showing a whitecap. At sea states of 4 (windspeeds of 11-16 knots), numerous whitecaps appear, and at sea states of 5 (windspeeds of 17-21 knots), many whitecaps form and spray is visible. When the sea state reaches 6 (windspeeds of 22-27 knots), whitecaps are uniformly everywhere and more spray is evident. The surface as a whole becomes darker as the windspeed increases; the waves are on the average darker, because they are larger and their faces steeper. Referring again to the reflection/absorption relationship, it may be seen that a rougher sea surface must be darker, because incidence angles relative to an observer near the surface are higher, causing less reflection and more absorption of incident light. A closer examination of the surface will also reveal small capillary waves. These are of short wavelengths (1-2 inches) and are caused predominantly by surface tension effects which are highly dependent upon the composition of water at the surface.

The Significance of the Sea Surface for Visual Cueing Purposes

Since the sea surface phenomenon just described is complex, it is appropriate to discuss the significance for visual cueing before suggesting ways to model it for computer image generation. Because the waves show darker faces parallel to the horizon, they offer pitch and roll cues. Their general darkening for locations approaching the nadir could provide an additional cue to pitch attitude, even if the horizon were not clearly visible. The dynamic nature of waves, i.e., their rise and fall, could provide a visual cue to absolute range to a wave provided that a large lifetime of a wave can be observed. Small waves have a short lifetime (1-2 seconds), while large waves have lifetimes of several seconds. The amount of a wave's lifetime that can be observed will depend on the wave size and the overflight speed and altitude. As sea states increase beyond 3, wave motion is more in the direction of the wind, wave crests begin to break, and foam and spray appear. These effects increase in intensity as the sea state increases further. Finally, small capillary waves dimple the large waves offering more cues to wave size and shape. While no strong cues to azimuth are present, when strong light sources such as the sun and moon are visible, their mottled reflection in the surface offer a strong cue of azimuth relative to the azimuth of these bodies. Some photographs of the sea surface are contained in Ref. 7.

Whether the flight task is to overfly the surface at low altitude, ride on it, for example, as when in a hovercraft or boat, and/or identify the sea state and windspeed, it is speculated that absolute attitude, heading and range cues are offered by the sea surface, as well as cues as to its state and prevailing windspeed and wind direction.

Suggested Initial Model for Computer Image Generation

All of the above suggests the use of a dynamic two-dimensional texture model for the sea surface that is predicated on wave statistics. Although the sea surface is really a three-dimensional one, its appearance is such that a two-dimensional rendition might suffice for oblique viewing at low altitude where the wave crests always appear below the horizon. It is

suggested that a single dynamic texture map representing a single wave be configured. This effect might be borrowed from that representing an explosion. The wave should be posted on a flat surface plane and caused to grow, increase in contrast, recede and fade away.

Our first attempt at creating an ocean appearance model is shown in an accompanying videotape. In it, we demonstrate the results of our first generation algorithms showing the shape, size and dynamic characteristics of several typical waves. We were limited to the representation of only a few waves, and we suspect that we need to decorate an immediate field with several hundred waves before the full effect would be apparent.

References

1. Sinacori, John B., "Research and Analysis of Head-Directed Area-of-Interest Visual System Concepts," prepared under Contract NAS2-10934 for NASA Ames Research Center, National Aeronautics and Space Administration, Aug., 1980.
2. Horowitz, Milton W., "James J. Gibson on Visual Perception: Analysis of Selected Papers," February 1964, Educational Research Corporation, Cambridge, Massachusetts. Technical Report No. NAVTRADEVCEEN 294-5, prepared for U.S. Naval Training Device Center, Port Washington, New York.
3. Zacharias, G.L., A. K. Caglayan, and P. M. Godiwala, "Design and Performance Analysis for An Image-Flow Terrain-Following Guidance System," July 1985. Technical Report No. R8501 prepared for J. B. Sinacori Associates under USAF ASD Contract No. F33615-81-0515.
4. Zacharias, G. L., "Modelling the Pilot's Use of Flight Simulator Visual Cues in a Terrain-Following Task," October 1985. Technical Report No. R8505 prepared for J. B. Sinacori Associates under USAF ASD Contract No. F33615-81-0515.
5. Minnaert, M., The Nature of Light & Color in the Open Air, Dover Publications Inc., 1954.

6. Van Dorn, William G., *Oceanography and Seamanship*, Dodd, Mead & Company, New York, 1974.

7. Defense Mapping Agency, Hydrographic Center (DMAHC), *Mariner's Guide to Wind Speed (Beaufort Scale) Sea State*.

AUTHOR INDEX

- Abbey, G. 12-FST-1
 Advani, S. 15-FST-4, 47-FST-7
 Allen, L. 12-FST-1
 Anderson, M. 13-FST-2
 Anderson, R. 12-FST-1
 Anderson, S. 12-FST-1
 Bailey, R. 79-FST-14
 Belo, E. 63-FST-10
 Bezdek, W. 13-FST-2
 Bueth, S. 79-FST-14
 Butler, G. 64-FST-11
 Caleffi, T. 63-FST-10
 Cardullo, F. 15-FST-4
 Chetelat, M. 47-FST-7
 Conley, J. 32-FST-6
 Culp, R. 63-FST-10
 Danek, G. 49-FST-9
 Darling, D. 63-FST-10
 Deiss, W. 47-FST-7
 Deppe, P. 79-FST-14
 Devarajan, V. 31-FST-5
 Dunham, E. 31-FST-5
 Ellis, T. 64-FST-11
 Fadden, E. 47-FST-7
 Fehse, W. 65-FST-12
 Fought, D. 31-FST-5
 Fu, H-L. 63-FST-10
 Gander, D. 78-FST-13
 Garrood, S. 48-FST-8
 George, G. 15-FST-4
 Goldiez, B. 49-FST-9
 Goman, M. 65-FST-12
 Grosz, J. 15-FST-4
 Guruprasad, G. 64-FST-11
 Hosman, R. 48-FST-8
 Howe, R. 47-FST-7
 Illauer, D. 78-FST-13
 Katz, A. 13-FST-2
 Khatwa, R. 32-FST-6
 Khrantsovsky, A. 65-FST-12
 Knotts, L. 79-FST-14
 Kull Jr., F. 31-FST-5
 Lin, K-C. 49-FST-9, 63-FST-10, 64-FST-11
 Lisle, C. 64-FST-11
 Little, J. 14-FST-3, 65-FST-12
 Lu, P. 63-FST-10
 Mansur, H. 13-FST-2
 Markman, S. 79-FST-14
 Matusof, R. 15-FST-4
 Mazurin, U. 32-FST-6
 McArthur, D. 31-FST-5
 Miao, A. 15-FST-4
 Miller, R. 12-FST-1
 Montag, B. 31-FST-5
 Moody, L. 13-FST-2
 Murthy, V. 13-FST-2
 Neville, K. 65-FST-12
 Ng, H. 49-FST-9
 Nieuwpoort, T. 32-FST-6
 Ollerios, F. 12-FST-1
 Panda, R. 47-FST-7
 Pickerine, J. 64-FST-11
 Ponomarenko, V. 78-FST-13
 Predtechensky, A. 13-FST-2, 15-FST-4
 Reid, L. 48-FST-8
 Reinbachs, N. 64-FST-11
 Richard, J. 49-FST-9
 Roach, C. 14-FST-3
 Rodchenko, V. 13-FST-2, 15-FST-4
 Rourke, R. 49-FST-9
 Rysdyk, R. 15-FST-4
 Sansom, R. 63-FST-10
 Sarathy, S. 13-FST-2
 Sawler, R. 15-FST-4
 Schab, D. 13-FST-2
 Schamle, M. 13-FST-2
 Schroeder, J. 48-FST-8
 Sheppard, S. 47-FST-7
 Shibaev, V. 48-FST-8
 Shiner, R. 47-FST-7
 Siksik, D. 49-FST-9
 Sinacori, J. 15-FST-4
 Smith, M. 63-FST-10
 Stephens, T. 65-FST-12
 Stuever, R. 31-FST-5
 Stypakov, G. 32-FST-6
 Sullivan, B. 47-FST-7
 Sweeney, C. 47-FST-7
 Tatarnikov, K. 65-FST-12
 Theunissen, E. 14-FST-3, 32-FST-6
 Tobias, A. 65-FST-12
 Ussov, V. 78-FST-13
 van der Steen, H. 48-FST-8
 van der Vaart, J. 15-FST-4
 van Hengst, J. 32-FST-6
 Vasilets, V. 14-FST-3, 47-FST-7
 Verspay, J. 32-FST-6
 Victory, D. 31-FST-5
 Vorona, A. 78-FST-13
 Wang, K. 14-FST-3
 Warren, R. 15-FST-4
 Welter, J. 64-FST-11
 Weske, R. 49-FST-9
 Wilkins, D. 14-FST-3
 Xueqiao, H. 63-FST-10
 Yakimenko, O. 47-FST-7
 Yakovlev, A. 14-FST-3
 Yashin, Y. 13-FST-2
 Zacharias, G. 15-FST-4
 Zaychik, L. 13-FST-2
 Zhang, B. 65-FST-12
 Zhou, P-N. 64-FST-11

